

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Disertační práce

2017

Ing. Jan Škach

Západočeská univerzita v Plzni
Fakulta aplikovaných věd

DETEKCE ZMĚN A CHYB V DYNAMICKÝCH SYSTÉMECH

Ing. Jan Škach

disertační práce
k získání akademického titulu doktor
v oboru Kybernetika

Školitel: doc. Ing. Ondřej Straka, Ph.D.
Katedra: Katedra kybernetiky

Plzeň, 2017

University of West Bohemia
Faculty of Applied Sciences

CHANGE AND FAULT DETECTION IN DYNAMIC SYSTEMS

Ing. Jan Škach

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy
in
Cybernetics

Advisor: doc. Ing. Ondřej Straka, Ph.D.
Department: Department of Cybernetics

Pilsen, 2017

PROHLÁŠENÍ

Prohlašuji, že jsem předloženou práci vypracoval samostatně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne

podpis

Poděkování

Na prvním místě patří poděkování Ing. Ivovi Punčochářovi, Ph.D., který se významně podílel svými nápady a cennými připomínkami na směr mého doktorského výzkumu. Dále bych rád poděkoval doc. Ing. Ondřejovi Strakovi, Ph.D. za celkovou podporu, odborné vedení doktorského studia a za možnost zapojit se do výzkumných projektů a zúčastnit se odborných konferencí. V neposlední řadě patří poděkování kolegům z Výzkumného týmu pro identifikaci systémů a rozhodování (IDM) a kolegům z výzkumného týmu Advanced Controls and Sensors Group (ACS), speciálně pak prof. Franku L. Lewisovi za inspiraci a možnost vědecké spolupráce.

Jan Škach

Disertační práce vznikla za podpory Ministerstva školství, mládeže a tělovýchovy, projekt číslo LO1506, Grantové agentury České republiky, projekt číslo GA15-12068S, a Fulbrightova programu.

Anotace

Složitost technických systémů roste stejně tak jako důležitost diagnostiky jejich chyb, která může pomoci snížit ekonomické výdaje, zachovat kvalitu a zvýšit bezpečnost. Detekci chyb lze rozdělit na pasivní a aktivní podle způsobu interakce mezi systémem a detektorem. Pasivní detekce chyb představuje tradiční přístup, který je dobře rozvinutý a je běžně používaný v reálných aplikacích. V poslední době se začala věnovat pozornost aktivnímu přístupu k detekci chyb, který je zaměřen na zvyšování kvality detekce chyb pomocí vhodně navrženého vstupního signálu sledovaného systému. Většina současných metod aktivní detekce chyb je věnována úloze řešené na konečném časovém horizontu, kde monitorování systému probíhá pouze ve vybraném časovém intervalu. V některých aplikacích je však vhodnější provozovat aktivní detektor chyb nepřetržitě na nekonečném horizontu.

Stěžejní část disertační práce je věnována rozvoji úlohy aktivní detekce chyb řešené na nekonečném časovém horizontu. Práce zároveň krátce pojednává o metodách pasivní detekce změn a chyb založených na datech, které lze využít v případech, kdy model systému není dopředu znám. Mezi hlavní cíle disertační práce patří formulace a návrh řešení úlohy aktivní detekce chyb pro diskrétní stochastické dynamické systémy na nekonečném časovém horizontu. Souběžně práce cílí na rozšíření formulace a řešení této úlohy o požadavky na chování systému.

Úloha aktivní detekce chyb a řízení je formulována jako optimalizační úloha na nekonečném časovém horizontu s obecným diskontovaným aditivním návrhovým kritériem. K popisu chování systému je použit vícemodelový přístup se známými pravděpodobnostmi přechodu mezi jednotlivými módy systému. Návrh aktivního detektoru chyb je rozložen do dvou kroků. Nejprve je původní úloha transformována na úlohu s úplnou informací o stavu využitím metod odhadu stavu. Následně se k řešení transformované úlohy využije aproximativní dynamické programování a zpětnovazební učení. Detekce změn a chyb bez znalosti modelu využívá geometrické vlastnosti lineárních systémů a učení bez učitele.

Annotation

The complexity of technical systems grows as well as the importance of fault diagnosis that can help to reduce economic costs, maintain quality, and increase safety. Fault detection can be divided into passive and active fault detection depending on the interaction between a system and a detector. Passive fault detection is a traditional approach that is well-developed and commonly used in real-world applications. Recently, attention has been aimed at the active approach that enhances the quality of fault detection using a suitably designed input signal that enters the monitored system. Most of the current methods to active fault detection deal with a finite time horizon problem where the system monitoring is performed only at a selected time interval. In some applications, however, it is preferable to run the active fault detector continuously on an infinite time horizon.

The main part of the dissertation is devoted to a development of the active fault detection on the infinite time horizon. Simultaneously, the thesis deals briefly with data-driven passive fault detection methods that can be used when the system model is not known in advance. The main objectives of the dissertation are the active fault detection problem formulation and its solution for discrete-time stochastic dynamic systems on the infinite time horizon. At the same time, the thesis aims to extend the formulation and the solution to this problem by integration of system behavior requirements.

The active fault detection problem is formulated as an optimization problem over the infinite time horizon with a general detection additive design criterion. Multiple-model framework with known transition probabilities between different modes is considered to describe the system behavior. Active fault detection design is decomposed into two steps. First, the original problem is transformed into a perfect state information problem by using state estimation methods. Consequently, approximate dynamic programming and reinforcement learning are used to solve the transformed problem. Data-driven change and fault detection without knowledge of a model uses geometric properties of linear systems and unsupervised learning.

Značení

Seznam vybraných anglicko-českých ekvivalentů

Některé z použitých odborných výrazů nejsou v české literatuře běžné. Z důvodu přehlednosti je vhodné zavést spojení vybraných anglických termínů s českými ekvivalenty. První sloupec následující tabulky obsahuje anglický výraz, ve druhém sloupci je uveden český ekvivalent.

Anglický název	Český překlad
ϵ -greedy policy	ϵ -hladová strategie
Active fault detection	Aktivní detekce chyb
Actor-critic structure	Struktura herec-kritik
Approximate dynamic programming	Aproximační dynamické programování
Artificial intelligence	Umělá inteligence
Bank of Kalman filters	Banka Kalmanových filtrů
Bayes risk	Bayesovo riziko
Bayesian recursive relations	Bayesovy rekurzivní vztahy
Behavior policy	Strategie chování
Bootstrap method	Metoda bootstrap
Curse of dimensionality	Prokletí rozměrnosti
Cyber-physical systems	Kyberneticko-fyzikální systémy
Data-driven methods	Metody založené na datech
Discount factor	Diskontní faktor
Eligibility vector	Vektor zodpovědnosti
Energy bounded uncertainty	Energeticky ohraničená nejistota
Excitation signal	Budicí signál
Fault signature matrix	Matice vlivu chyb
Fault-tolerant Control	Řízení odolné proti chybám
Generalized Pseudo Bayes algorithm	Zobecněný pseudo Bayesův algoritmus
Growing cell structure	Rostoucí buněčná struktura
Growing neural gas	Rostoucí neuronový plyn
Hierarchical aggregation	Hierarchická agregace

Anglický název	Český překlad
Informative statistics	Informační statistiky
Interacting multiple model algorithm	Interagující vícemodelový algoritmus
Institute of Electrical and Electronics Engineers	Institut pro elektrotechnické a elektronické inženýrství
International Federation of Automatic Control	Mezinárodní federace automatického řízení
Internet of things	Internet věcí
Kernel function	Jádrová funkce
Large-scale systems	Vícerozměrové systémy
Moment matching method	Metoda slučování
National Aeronautics and Space Administration	Národní úřad pro letectví a kosmonautiku
Off-policy learning	Off-policy učení
Particle filter	Částicový filtr
Pattern classification	Rozpoznávání vzorů
Pointwise bounded uncertainty	Bodově ohraničená nejistota
Policy evaluation	Ohodnocení strategie
Policy improvement	Vylepšení strategie
Policy iteration	Iterace strategie
Principal component analysis	Analýza hlavních komponent
Optimal experimental design	Optimální návrh experimentů
Q-function	Q-funkce
Quantization	Kvantizace
Radial basis function	Radiální bázová funkce
Reinforcement learning	Zpětnovazební učení
Russian Federal Space Agency	Ruská Federální kosmická agentura
Self-organizing map	Samoorganizující se mapa
Sequential probability ratio test	Sekvenční test poměru věrohodností
Sequential quadratic programming	Sekvenční kvadratické programování
Singular value decomposition	Singulární rozklad
Sparse matrix	Řídká matice
Successive approximation	Postupná aproximace
Sufficient statistics	Postačující statistiky
Supervised learning	Učení s učitelem
Target policy	Cílová strategie
Temporal difference error	Chyba temporální diference
Temporal difference learning	Učení temporální diferencí
Temporal difference Q-learning	Q-učení temporální diferencí

Anglický název	Český překlad
Unscented transform	Unscentovaná transformace
Unsupervised learning	Učení bez učitele
Value function	Užitková funkce
Value iteration	Iterace užitkové funkce
Youla–Jabr–Bongiorno–Kucera parametrization	Youla-Kučerova parametrizace

Seznam zkratek

Pro přehlednost jsou v následující tabulce uvedeny zkratky, které se vyskytují v této práci. První sloupec tabulky obsahuje zkratku a druhý sloupec uvádí ekvivalentní český termín. V samotném textu práce je zkratka před prvním použitím zadefinována pro lepší návaznost.

Zkratka	Celý název
BRV	Bayesovy rekurzivní vztahy
EKF	Rozšířený Kalmanův filtr
ZPB	Zobecněný pseudo Bayesův algoritmus
CHTD	Chyba temporální diference
i.i.d.	Nezávislé a stejně rozdělené (náhodné proměnné)
IEEE	Institut pro elektrotechnické a elektronické inženýrství
IFAC	Mezinárodní federace automatického řízení
KF	Kalmanův filtr
LQR	Lineární kvadraticky optimální řízení
MC	Metoda Monte Carlo
QTD	Q-učení temporální diferencí
TD(λ)	Temporální diference s parametrem λ

Obsah

1	Úvod	3
2	Detekce chyb	5
2.1	Úloha detekce chyb	5
2.2	Aktivní detekce chyb	7
2.2.1	Pravděpodobnostní přístup	9
2.2.2	Deterministický přístup	11
2.3	Optimální aktivní detekce chyb a řízení	13
2.3.1	Formulace úlohy	14
2.3.2	Návrh řešení	16
3	Cíle disertační práce	17
3.1	Motivace	17
3.2	Cíle	18
3.2.1	Optimální aktivní detekce chyb na nekonečném časovém horizontu	18
3.2.2	Detekce změn a chyb bez znalosti modelu	19
4	Formulace úlohy aktivní detekce chyb na nekonečném časovém horizontu	20
4.1	Formulace úlohy	20
4.1.1	Model systému	20
4.1.2	Aktivní detektor chyb	22
4.1.3	Návrhové kritérium	22
4.2	Úloha s neúplnou informací o stavu	23
4.2.1	Postačující statistiky	23
4.3	Aktivní detekce chyb a řízení	25
5	Přeformulování úlohy	27
5.1	Informační statistiky	27
5.1.1	Úloha s přímo měřitelnou spojitou částí stavu	28
5.1.2	Lineární gaussovské systémy s nepřímo měřitelnou spojitou částí stavu	30
5.1.3	Nelineární gaussovské systémy s nepřímo měřitelnou spojitou částí stavu	34

5.1.4	Nelineární negaussovské systémy s nepřímo měřitelnou spojitou částí stavu	35
5.2	Postup přeformulování úlohy	37
5.2.1	Úloha s přímo měřitelnou spojitou částí stavu	39
5.2.2	Úloha s nepřímo měřitelnou spojitou částí stavu	41
6	Řešení optimalizační úlohy s úplnou informací o stavu	44
6.1	Dynamické programování	44
6.2	Transformace modelu na Markovův rozhodovací proces	48
6.2.1	Přeformulování úlohy	48
6.2.2	Řešení dynamickým programováním	50
6.2.3	Adaptivní algoritmus zobecněné iterace strategie	53
6.3	Přímá kvantizace spojitého stavového prostoru	54
6.4	Učení temporální diferencí pomocí lineární parametrické aproximace	58
6.4.1	Lineární parametrická aproximace	58
6.4.2	Učení temporální diferencí	61
6.4.3	Q-učení temporální diferencí	63
6.5	Použití neparametrické aproximace	66
6.6	Dynamické programování pro úlohu aktivní detekce chyb a řízení	69
7	Detekce změn a chyb bez znalosti modelu	70
7.1	Detekce změn a chyb založená na datech	70
7.2	Detekce změn využívající jádra lineárního zobrazení	72
7.3	Detekce změn využívající samoorganizující se mapy	74
7.4	Využití detekce změn v úloze optimálního řízení bez znalosti modelu	77
8	Ilustrační příklady	80
8.1	Aktivní detekce chyb a řízení	80
8.1.1	Chemostat	80
8.1.2	Kyvadlo	88
8.1.3	Mobilní robot	97
8.2	Detekce změn a chyb bez znalosti modelu	105
8.2.1	Systém prvního řádu	105
9	Závěr	109
9.1	Shrnutí dosažených výsledků	109
9.2	Možnosti budoucího výzkumu	113

Kapitola 1

Úvod

S mohutným rozšířením automatizace v celé řadě oblastí roste i počet aplikací, které jsou bezpečnostně kritické. Vzhledem k tomu, že lidské schopnosti odhalení skrytých chyb jsou omezené, nové metody automatického rozpoznání změn a chyb v systému tvoří velmi důležitý směr současného výzkumu. Jejich využití lze najít v různých aplikacích jako jsou autonomní vozidla, letadla, inteligentní budovy nebo například výrobní a chemické závody. Nechtěné změny chování systému mohou nastat v důsledku vnějších nebo vnitřních vlivů. Vysoká vlhkost a změny teploty mohou vést ke korozi. Nedostatečné mazání může být příčinou vyššího tření, v jehož důsledku dochází k přehřívání nebo prasklinám. Všechny tyto druhy degradace senzorů, akčních členů a systémových komponent jsou přirozené a pokud není učiněna příslušná náprava, mohou vést ke katastrofickým událostem, které mohou způsobit ekonomické ztráty, ekologické znečištění nebo dokonce ztráty lidských životů.

Aby se zabránilo těmto katastrofickým událostem, jsou do sledovaných systémů implementovány detektory monitorující změny chování. Tato zařízení mohou plnit svá poslání použitím nejrůznějších technologií a postupů. Běžně používaným přístupem sloužícím k odhalení nežádoucího chování je pasivní detekce chyb [1], [2], kdy rozhodnutí o chybě je generováno na základě vstupně výstupních dat bez jakékoliv interakce mezi detektorem chyb a sledovaným systémem. Druhým přístupem je aktivní detekce chyb založená na vzájemném působení detektoru chyb a systému [3], [4]. Jedná se o atraktivní a perspektivní přístup k návrhu detektoru chyb, jehož klíčová myšlenka spočívá v aplikaci vhodně navrženého budicího vstupního signálu na vstup sledovaného systému. Tento signál je navržen tak, aby byla zvýšena kvalita rozhodnutí o chybě. Návrh vhodného budicího signálu byl již dříve diskutován v jiných technických oblastech. Vhodně navržený vstupní signál se využívá ke zlepšení poznávání zkoumaného systému v oblasti identifikace systémů [5], ke konstrukci experimentů v oblasti optimálního návrhu experimentů [6], nebo ke zvýšení kvality řízení v oblasti duálního řízení [7].

Při zjišťování nežádoucího chování systému mohou hrát roli různé faktory. Některé aplikace mohou vyžadovat zaručení správného odhalení chyby za konečný časový horizont. V jiných situacích je důležitá flexibilita návrhových požadavků. Další úlohy cílí kromě detekce chyb i na řízení systémů. V současné době existuje mnoho metod aktivní detekce chyb, které předpokládají buď omezené nejistoty v systému [4], [8]–[12] nebo nejistoty popsané

pravděpodobnostními rozděleními [3], [13]–[16]. Většina těchto metod předpokládá návrh budícího vstupního signálu na konečném časovém horizontu. Z důvodu zvýšené složitosti a výpočetní náročnosti se takové metody nehodí pro úlohy, kdy je zapotřebí kontinuálně monitorovat systém. Proto je do disertační práce promítnuta snaha o návrh aktivního detektoru chyb, který zohledňuje obecně zvolené požadavky na kvalitu detekce chyb na nekonečném časovém horizontu a přitom respektuje nejistotu v systému. Formulace úlohy detekce chyb může být zároveň jednoduše rozšířena o požadavky na chování systému a vyústit v úlohu návrhu optimálního aktivního detektoru chyb a regulátoru.

Velkou skupinu metod detekce změn a chyb tvoří vícemodelové metody, které jsou na základě dat schopny rozlišit různé módy chování systému. Jednotlivé módy systému pak mohou být reprezentovány modelem chování. Někdy však může být samotné sestavení modelů časově i finančně náročné. Především v situacích, kdy je k dispozici velké množství nasbíraných dat, lze využít metody analyzující měřená data. Kromě aktivní detekce chyb se disertační práce v krátkosti věnuje právě návrhu algoritmů, které jsou schopny bez znalosti modelu systému detekovat změnu v chování pouze na základě rozboru dostupných dat. Algoritmy jsou inspirované geometrickými vlastnostmi lineárních systémů a učením bez učitele [17].

Celá práce je rozdělena do devíti kapitol s logickou návazností. Po úvodní kapitole následuje kapitola 2, která se zabývá představením úlohy detekce chyb a obecným popisem pasivního a aktivního přístupu. Aktivní přístup je dále rozdělen do dvou hlavních směrů. Kapitola 3 definuje cíle disertační práce a jejich motivaci. V následující kapitole 4 je formulována optimalizační úloha návrhu aktivního detektoru chyb na základě minimalizace obecného návrhového kritéria na nekonečném časovém horizontu. Jelikož řešení formulované úlohy s neúplnou informací o stavu je složité a výpočetně náročnější než v případě úlohy s úplnou informací o stavu, předmětem kapitoly 5 je vhodné přeformulování takové úlohy. Různé postupy hledání řešení přeformulované úlohy jsou představeny v kapitole 6. Nejprve je uvedeno obecné řešení využívající dynamické programování. Jelikož v případě formulované úlohy nemůže být řešení nalezeno analyticky, kapitola 6 se zároveň zabývá různými postupy k nalezení suboptimálního řešení. V kapitolách 4, 5 a 6 je kromě úlohy aktivní detekce chyb zároveň krátce diskutována úloha aktivní detekce chyb a řízení. Detekci změn a chyb bez znalosti modelu se věnuje kapitola 7, ve které je zároveň stručně diskutováno použití detekce změn v úloze optimálního řízení. Kapitola 8 obsahuje numerické příklady, které slouží jako ukázka správné funkce navržených algoritmů. Závěrečná kapitola 9 shrnuje dosažené výsledky a představuje potencionální budoucí směry výzkumu.

Kapitola 2

Detekce chyb

Cílem této kapitoly je vymezit oblast detekce chyb, kterou se disertační práce zabývá. Podkapitola 2.1 představující obecný úvod do oblasti detekce chyb obsahuje motivaci na základě reálných případů událostí spojených s nedetekováním chyby, přehled vybraných pojmů a obecný přehled přístupů k detekci chyb. Přístupy jsou v následující podkapitole 2.2 klasifikovány na základě interakce mezi systémem a detektorem chyb do dvou skupin, jako pasivní a aktivní. V textu jsou následně uvedeny charakteristiky aktivního přístupu a rozdělení metod v závislosti na popisu neurčitostí v systému. Podkapitola 2.3 pak popisuje pravděpodobnostní metodu detekce chyb a řízení systému nazvanou jako optimální aktivní detekce chyb a řízení, která je předmětem zájmu disertační práce.

2.1 Úloha detekce chyb

S rostoucí složitostí systémů se zvyšuje důraz na bezpečnost jejich chodu. Včasné rozpoznaní nežádoucího chování systému může předejít nežádoucím důsledkům, jako jsou finanční ztráty související s chybným chováním, způsobení přírodní katastrofy nebo dokonce ztráta lidských životů. Příkladem mohou být události z nedávné doby. Evropská vesmírná agentura a ruská Federální kosmická agentura se musejí vypořádat s následky ztráty vesmírného modulu Schiaparelli "vstup – sestup – přistání" sondy ExoMars. Když se modul v říjnu 2016 pokusil o přistání na Marsu, vysílaný signál byl ztracen a jeho trosky byly později identifikovány orbitální sondou Národního úřadu pro letectví a kosmonautiku. Příčinou nehody byla chybná informace o výšce modulu před přistáním, v jejímž důsledku byla sonda předčasně odtržena od padáku [18]. Pád byl navíc umocněn chybným zážehem raketových motorů, který byl příliš krátký. Kromě vynaložených ztrát ovlivní tato událost zcela jistě i další průběh mise ExoMars zahrnující výzkum povrchu planety. Jiná významná událost ovlivnila životy deseti tisíců lidí po celé planetě a negativně ovlivnila pověst korejské společnosti Samsung. Výrobní vada a nedostatečná schopnost mobilního telefonu upozornit na chybné chování způsobily, že korejská společnost musela záhy po začátku prodeje modelu Note 7 telefon celosvětově stáhnout. Příčinou výbuchů byla kombinace vnitřního uspořádání baterie spolu s deformací baterie v jejím rohu způsobená konstrukcí telefonu [19].

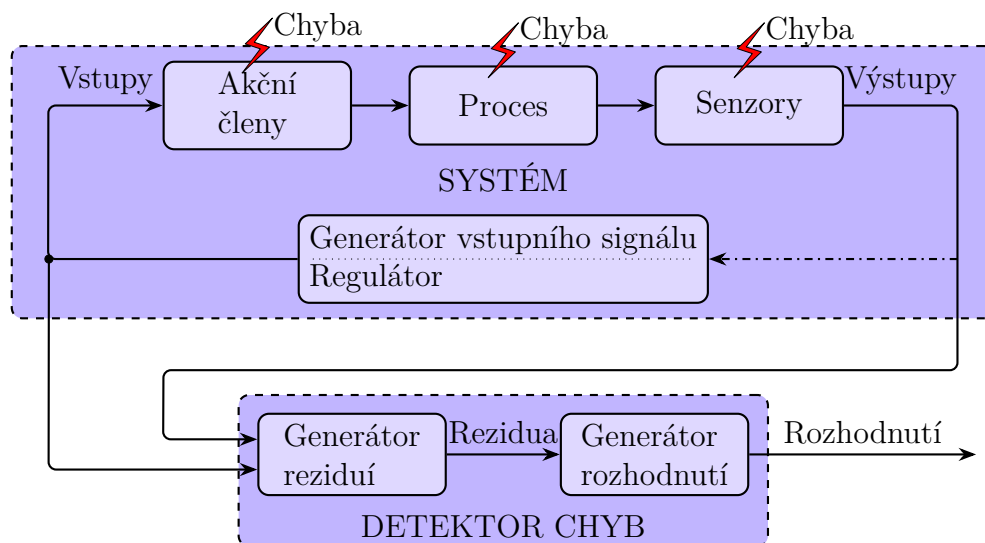
Mezi časté nehody, které mají dopad na životní prostředí a zároveň přímo ohrožují lidské životy patří havárie v chemických závodech a rafineriích. Na konci roku 2016 se jednalo o požár v rafinerii společnosti Exxon Mobil v americké Louisianě nebo v rafinerii italské společnosti Eni blízko městečka Sannazzaro de' Burgondi na severu Itálie. Detekce chyb si proto zaslouží zvýšenou pozornost, aby se omezil výskyt podobných událostí.

V případě řízeného systému mohou být malé poruchy nebo pozevolné změny kompenzovány standardními nebo adaptivními regulátory. Nicméně některé změny a anomálie nelze tímto způsobem dostatečně kompenzovat. Pojem detekce chyb souvisí se schopností odhalit existenci chyby v systému a její identifikaci, přičemž *chybou* v dynamickém systému se obvykle myslí odchylka struktury systému nebo parametrů systému z jejich běžných bezchybných hodnot [20]. Struktura systému se může změnit v důsledku odpojení aktuátorů a senzorů od systému, případně odpojení některých dalších částí systému. Změny parametrů mohou být výsledkem změny vlastností částí systému. Jelikož je oblast detekce chyb rozšířena i v dalších technologických oblastech, je zapotřebí jasně definovat vybrané odborné názvy. Přehledná terminologie detekce chyb sestavená odbornou komisí Mezinárodní federace automatického řízení (IFAC-Technical Committee), zabývající se detekcí chyb, dohledem a bezpečností technických procesů, je dostupná například v [21]. Termíny použité v této práci se drží zmíněné terminologie kromě definice pojmu *detekce chyb*, který zde zahrnuje jak schopnost určení přítomnosti chyby a času detekce, tak její izolaci. Mezi základní požadavky metod detekce chyb patří včasná detekce náhlých změn chování systému, rozpoznání příčin vzniku chyby, nebo detekce chyby ve zpětnovazební smyčce. V důsledku správně navrženého detektoru chyb by se měl systém stát bezpečným pro své okolí a měl by být schopný dodržet požadavky na správnou funkci za daných podmínek.

V oblasti detekce chyb existuje celá řada přístupů, které byly navrženy v posledních několika desetiletích. Jeden z historicky nejstarších přístupů je založen na redundanci senzorů, kdy veličina je měřena třemi a více senzory. V případě, že jeden senzor naměří hodnotu měřené veličiny zřetelně rozdílnou od hodnoty poskytnuté zbývajícími senzory, je tento senzor označen za chybný. Jedná se o velmi jednoduchý a elegantní přístup, který má však i své nedostatky, a to vyšší nároky na cenu, hmotnost nebo prostor. Novější přístupy k detekci chyb se dají rozdělit do tří skupin: signálové, modelové a přístupy založené na datech [22]. Signálové metody se opírají o analýzu signálů v časové nebo frekvenční oblasti [1]. V případě časové oblasti se jedná zejména o metody založené na kontrole daných mezi nebo pozorování průběhu měřeného signálu v čase. Dále mohou být zkoumány statistické vlastnosti jako je průměrná hodnota nebo výběrová variance. Metody zabývající se analýzou signálů ve frekvenční oblasti mohou vyhodnotit chybné chování systému například na základě porovnání frekvenčního spektra v případě bezchybného a chybného chování systému. Modelové metody předpokládají znalost modelu chování systému, na jehož základě jsou generována rozhodnutí o chybě [23], [24]. Model chování systému může být získán matematicko-fyzikálním modelováním [25], nebo identifikací z reálných dat [5]. V prvním případě se předpokládá, že se chyba systému projeví například změnou parametrů modelu. Pomocí odhadu parametrů, odhadu stavu nebo paritními rovnicemi lze rozhodnout o chybě v systému. Druhá skupina modelových metod založená na identifikaci předpokládá možnost trénování zvolené struktury modelu pomocí reálných dat. Model následně poslouží pro

návrh detektoru chyb. S rostoucím množstvím senzorů se zároveň zvyšuje množství naměřených dat, které musí být zpracováno. Přístup k detekci chyb založený na datech cílí na nalezení podstatné informace z množství dat a stanovení vztahu mezi daty a chybami [2], [26], [27]. K řešení obtížných problémů reálného světa využívají metody založené na datech umělé inteligence a mezi nejčastější reprezentace modelu patří neuronové sítě a fuzzy logika [28]. Existuje celá řada odborných zdrojů mapujících různé přístupy k detekci chyb [20], [29]–[32], zatímco další odborné zdroje zkoumají možnost jejich aplikace [21], [33]–[36].

Základní schéma pro detekci chyb je znázorněno na obrázku 2.1. Systém se skládá ze samotného procesu a dále z akčních členů, senzorů a generátoru vstupního signálu. V případě požadavků na chování systému s využitím zpětné vazby vstupují výstupy ze senzorů do regulátoru, jak je ilustrováno čerchovanou čarou na obrázku 2.1. Regulátor následně generuje příslušné řízení. Pro odhalení změn chování v systému se využívá *detektor chyb*, který je tvořen generátorem reziduí a generátorem rozhodnutí. *Reziduum* je indikátor poruchy, který porovnává předpokládané chování signálů s měřenými hodnotami. Reziduum významně změní své chování v případě chybného chování systému. *Generátor rozhodnutí* vyhodnotí rezidua a generuje rozhodnutí o chybě v systému, případně indikuje, o jakou chybu se jedná.

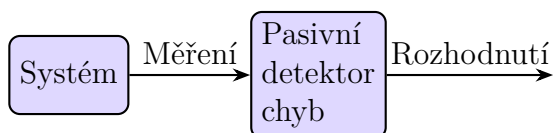


Obrázek 2.1: Základní schéma pro detekci chyb.

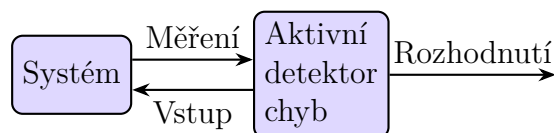
2.2 Aktivní detekce chyb

Detekci chyb lze rozdělit na dva přístupy podle způsobu interakce mezi systémem a detektorem chyb [37], [38]. Tradiční přístup je označován jako *pasivní detekce chyb*, jelikož nedochází k vzájemnému působení mezi systémem a detektorem chyb. To je zobrazeno na obrázku 2.2, který představuje základní blokový diagram pasivního přístupu. Pasivní přístup je teoreticky dobře vyvinutý, zavedený, úspěšně aplikovaný na různé reálné problémy

a ucelený přehled různých metod může být nalezen například v [31], [39], [40]. Může se stát, že chyby projevující se ve vstupně-výstupních datech pouze nepatrně nemusí být odhaleny, a nebo mohou být odhaleny až za nepřijatelně dlouhou dobu. Vědecký výzkum vyústil v inovativní koncept, tedy aktivní přístup k detekci chyb.

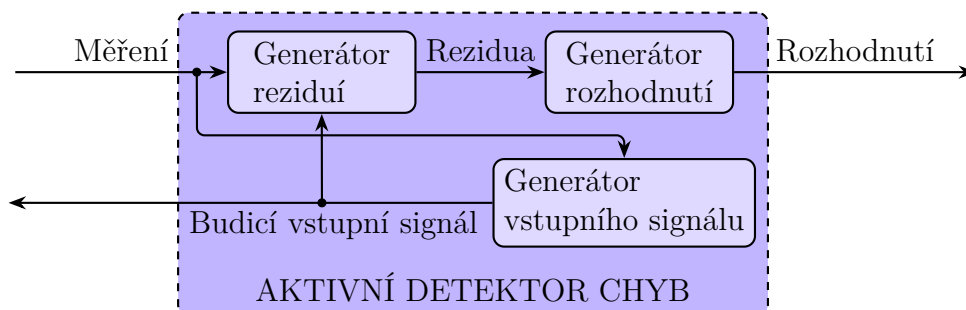


Obrázek 2.2: Základní schéma pasivní detekce chyb.



Obrázek 2.3: Základní schéma aktivní detekce chyb.

Aktivní detekce chyb je založena na myšlence zlepšení kvality detekce chyb využitím vhodně navrženého vstupního signálu. Tento signál má budicí charakter a jeho úkolem je umožnit získání co nejvíce informace o systému. V důsledku bohatší informace o sledovaném systému mohou být odhaleny chyby, které by v případě pasivní detekce chyb zůstaly skryté. Aktivní detektor chyb bývá navržen tak, aby kvalita detekce, měřená pravděpodobností chybného rozhodnutí, rychlostí detekce, případně jinými ukazateli, byla zvýšena. Při porovnání obrázků 2.2 a 2.3 je zřejmé, že v případě aktivní detekce chyb dochází k oboustranné komunikaci mezi systémem a detektorem chyb prostřednictvím signálu generovaného detektorem chyb. Vnitřní struktura aktivního detektoru chyb je znázorněna na obrázku 2.4. Na základě vstupů a výstupů systému jsou generátorem reziduí vytvořena rezidua a generátorem vstupního signálu budicí signál, který je vstupem do systému. Rozhodnutí jsou generována na základě reziduí v generátoru rozhodnutí. Naprostá většina metod aktivní detekce chyb je založena na vícemodelovém přístupu k detekci chyb. *Vícemodelový přístup* předpokládá, že chování systému může být popsáno několika modely, přičemž jeden model popisuje bezchybné chování systému a zbylé modely popisují případ, kdy v systému nastala chyba. Generátor rozhodnutí indikuje, který z modelů nejvíce odpovídá chování systému. Aktivní detekci chyb je možné dále rozdělit na dva přístupy: deterministický a pravděpodobnostní. Základní přehled metod obou přístupů je uveden v následující části, avšak podrobné srovnání vybraných metod může být nalezeno například v [Š10].



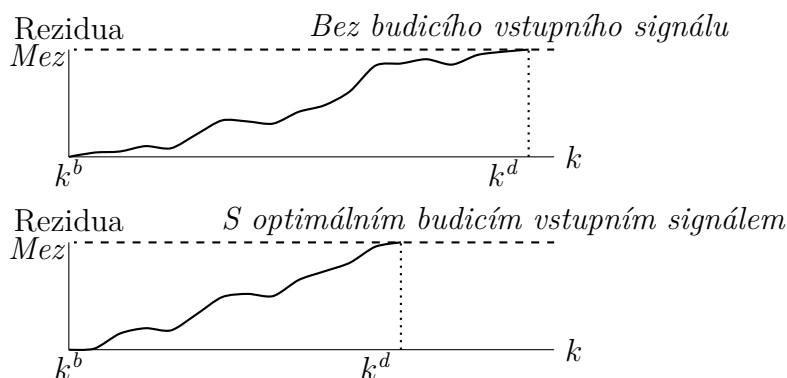
Obrázek 2.4: Vnitřní struktura aktivního detektoru chyb.

2.2.1 Pravděpodobnostní přístup

Ve reálném světě jsou systémy pod neustálým vlivem různých faktorů. Pravděpodobnostní přístup je tvořen metodami, které modelují nejistoty v systému jako náhodné procesy se známými pravděpodobnostními rozděleními. Pravděpodobnostní metody mohou být rozděleny na metody založené na statistickém testování hypotéz a metody využívající Bayesův přístup.

Metody založené na statistických testech

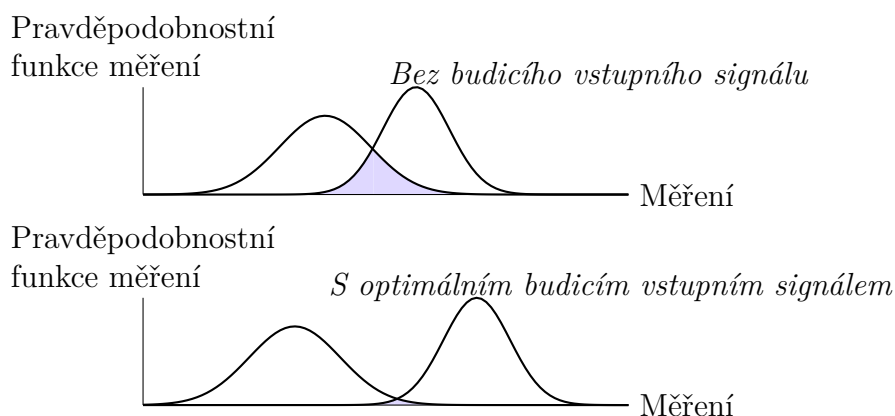
Jedna z prvních metod aktivní detekce chyb byla představena v [3]. Tato metoda předpokládá, že chování systému odpovídá buď bezchybnému modelu, nebo modelu představující chybné chování. Rozhodnutí o chybě jsou generována na základě sekvenčního testu poměru věrohodností [41]. Bylo ukázáno, že vhodně navržený budicí vstupní signál může snížit průměrný čas detekce chyb a pravděpodobnost falešných alarmů, neboli poměr chybných rozhodnutí vůči celkovému počtu rozhodnutí. Základní princip sekvenčního testu poměru věrohodností a porovnání pasivní s aktivní detekcí chyb je ilustrováno na obrázku 2.5. V každém časovém okamžiku $k \in \mathcal{T} = \{0, 1, \dots\}$ jsou generována rezidua na základě sekvenčního testu poměru věrohodností, do kterého se promítne měřený výstup systému. Chyba v systému je detekována v případě, že reziduální signál překročí pevně stanovenou mez. Necht $k^b \in \mathcal{T}$ je časový okamžik vzniku chyby a $k^d \in \mathcal{T}$ je časový okamžik detekce chyby. Z obrázku 2.5 je patrné, že v případě vhodně navrženého budicího vstupního signálu se čas detekce chyby zmenší. Původní návrh generátoru vstupního signálu byl založen na minimalizaci průměrného počtu časových okamžiků k dosažení rozhodnutí. Návrhové požadavky byly později rozšířeny o požadavek na minimalizaci počtu falešných alarmů [13].



Obrázek 2.5: Ilustrace vlivu vhodně navrženého budicího vstupního signálu, který cílí na zvýšení rychlosti detekce chyb pro metody založené na statistických testech. Měřítko obou grafů jsou stejná.

Metody založené na Bayesově přístupu

Bayesův přístup umožňuje přirozeně spojit měřená data a apriorní informaci. Podle Bayesovy věty může být apriorní informace aktualizována na aposteriorní s využitím měření. Základní myšlenka detekce chyb spočívá v minimalizaci pravděpodobnosti nesprávného rozhodnutí. Jelikož rozhodnutí je generováno na základě měřených dat, generátor budicího vstupního signálu se zaměřuje na ovlivnění věrohodnostních funkcí měřeného výstupu systému pro všechny možné modely chování systému. Tato myšlenka je ilustrována na obrázku 2.6, který zobrazuje pravděpodobnostní funkce měření dvou modelů chování systému. Vyšší funkční hodnoty představují větší věrohodnost daného modelu. Kvalita rozhodnutí o chybě je negativně ovlivněna, jestliže měřený výstup systému spadá do překrývajících se oblastí věrohodností měření. Jelikož mnoho běžně používaných pravděpodobnostních rozdělání má neomezený nosič funkce, je zde možnost nesprávného rozhodnutí z důvodu překrytí pravděpodobnostních funkcí měření. Vhodně navržený budicí vstupní signál může zajistit, že regiony s nenulovými hodnotami hustoty pravděpodobnosti měřeného výstupu jsou odděleny, jak je ilustrováno na obrázku 2.6. V důsledku zmenšení překrývajících se oblastí je kvalita detekce vylepšena. Existuje celá řada metod s různými návrhovými kritérii a různým řešením.



Obrázek 2.6: Ilustrace vlivu vhodně navrženého budicího vstupního signálu, který cílí na rozlišení hustot pravděpodobnosti měření pro různé modely chování systému.

Inspirací pro tvorbu návrhových kritérií slouží poznatky z nejrůznějších oblastí včetně oblasti zpracování signálu [42] nebo rozpoznávání vzorů [43]. Generátor rozhodnutí na základě měření rozhodne, který model je věrohodnější. Celkové riziko představují očekávané ztráty spojené s těmito rozhodnutími. V [44] je navržena metoda, která cílí na minimalizaci rizika. Jelikož tzv. Bayesovo riziko [43] nelze v dané úloze analyticky vyjádřit, autoři stanoví horní hranici Bayesova rizika. Metoda byla dále rozšířena pro systémy s libovolným počtem modelů chování [45] a pro vícenásobné změny chování [14]. Postup je navržen pro lineární diskrétní systémy na konečném časovém horizontu a řešení optimalizační úlohy využívá sekvenční kvadratické programování [46]. Podobná myšlenka návrhu vstupního signálu je uvedena v [16], kde autoři zkoumají míru podobnosti dvou pravděpodobnostních rozdělání,

kteřá je vyjádřena pomocí tzv. Hellingerovy vzdálenosti. Formulace úlohy umožňuje vykonávat aktivní detekci chyb pro nelineární systémy na konečném časovém horizontu. Obecnější pojetí návrhu vhodného vstupního signálu je představeno v [47]. Podobně jako v [13] se předpokládá, že generátor rozhodnutí je předem stanovený a cílem úlohy je rozhodnutí o chybě na konci uvažovaného časového horizontu detekce. Obecná myšlenka návrhu generátoru rozhodnutí a generátoru vstupního signálu založená na vhodně zvoleném návrhovém kritériu, které hodnotí kvalitu rozhodnutí pro konečný časový horizont, je uvedena v [48]. Ucelený rámec návrhu aktivního detektoru chyb pro diskrétní nelineární systémy s obecným kritériem je prezentován v [15]. K řešení optimalizační úlohy minimalizace kritéria se využívá dynamické programování [49], [50]. Tato metoda je blíže představena v podkapitole 2.3, jelikož jedním z cílů této práce je rozšíření této metody pro nekonečný časový horizont. Další rozšíření diskutované metody na konečném časovém horizontu zahrnuje zapracování omezujících podmínek, které mohou plynout z fyzikálních nebo jiných omezení systému. Ucelený popis metody aktivní detekce chyb založené na řešení optimalizační úlohy s omezeními je uveden v [51].

2.2.2 Deterministický přístup

Souběžně s pravděpodobnostním přístupem existuje druhý aktivní přístup, který předpokládá deterministický popis nejistot v systému. Na rozdíl od metod využívajících pravděpodobnostního popisu nejistot může být v případě omezených nejistot zaručena správná detekce chyb v konečném čase, jelikož množiny měřených výstupů mohou být odděleny pro jednotlivé modely chování systému. Deterministické metody mohou být dále rozděleny na množinové metody a metody využívající integrovaný návrh s použitím \mathcal{H}_2 a \mathcal{H}_∞ optimalizace.

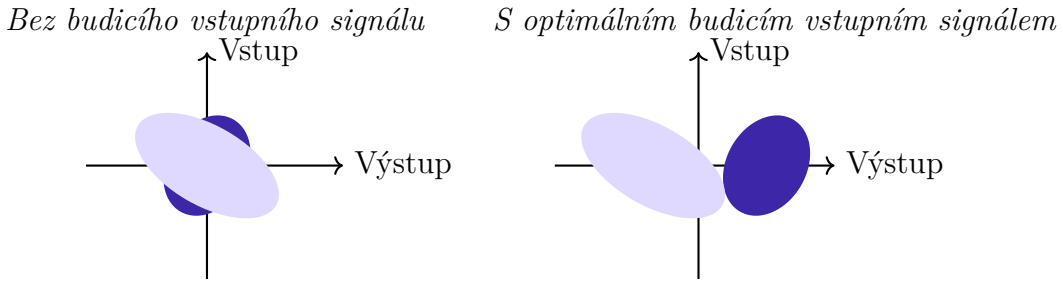
Metody využívající integrovaný \mathcal{H}_2 a \mathcal{H}_∞ návrh

Počátky deterministických metod jsou úzce spojeny s integrovaným přístupem k řízení a detekci chyb pomocí regulátoru se čtyřmi stupni volnosti [52]. Dvě přenosové funkce přísluší samotnému regulátoru a dvě přenosové funkce generátoru reziduí. Návrh regulátoru a generátoru reziduí tedy není prováděn nezávisle, ale je formulován jako jedna optimalizační úloha. Později bylo ukázáno [53], že regulátor se čtyřmi stupni volnosti představuje speciální případ obecnější struktury spojení mezi vstupy a výstupy používané v \mathcal{H}_2 a \mathcal{H}_∞ optimalizaci. V práci bylo konstatováno, že pro případ \mathcal{H}_2 indexu měření kvality musí být regulátor a generátor reziduí navrženy dohromady při výskytu nejistoty ve formě chyby modelování. Další rozvoj návrhu aktivního detektoru chyb a regulátoru je představen v [54], kde lze najít detailněji rozpracovanou metodiku návrhu s ohledem na robustnost systému využívající Youla-Kučerovu parametrizaci [55]. Smíšená \mathcal{H}_2 a \mathcal{H}_∞ formulace úlohy, kdy cíle řízení jsou měřeny pomocí \mathcal{H}_∞ normy a cíle detekce chyb pomocí \mathcal{H}_2 normy je uvažována v [56]. Další metoda aktivní detekce chyb a řízení pro lineární systémy založená na Youla-Kučerově parametrizaci zároveň definuje matici vlivu chyb sloužící k izolaci chyb [57], [58]. Pokud nelze docílit přímé izolace možných chování systému z měřených dat, je možné po-

užít dynamickou izolaci chyb pomocí optimálně navrženého budicího vstupního signálu. Ten je navržen tak, aby minimalizoval vliv na řízený výstup systému v případě bezchybného chování systému a zároveň maximalizoval vliv na rezidua v případě chybného chování systému. Využití Youla-Kučerovy parametrizace při návrhu aktivního detektoru chyb bylo dále diskutováno například v [59], [60].

Množinové metody

Stavebním kamenem množinových metod je předpoklad, že neurčitosti v systému mohou být popsány veličinami, které nabývají hodnot z předem známého omezeného rozsahu. Měřený výstupní signál systému jako odezva na vstupní signál náleží určité množině trajektorií. Budicí vstupní signál je navržen tak, aby byly množiny vstupně-výstupních trajektorií pro různé modely chování systému disjunktní. Tímto způsobem je možné zaručit správnou detekci chyb. Oddělení množin vstupně-výstupních trajektorií pomocí vhodně navrženého budicího vstupního signálu je ilustrováno na obrázku 2.7.



Obrázek 2.7: Ilustrace vlivu vhodně navrženého budicího vstupního signálu, který vede k disjunktním množinám vstupně-výstupních trajektorií.

Deterministické metody obvykle předpokládají energetické nebo bodové ohraničení nejistot. Energetické ohraničení představuje případ, kdy je součet hodnot druhých mocnin nejistot na celém časovém horizontu detekce chyb omezen konečnou mezí, zatímco v případě bodového ohraničení jsou nejistoty omezeny konečnou mezí v každém časovém okamžiku. Necht' pro každý časový okamžik k na konečném horizontu $k = 0, 1, \dots, F$ je definován stav $\mathbf{x}_k \in \mathbb{R}^{n_x}$ a celková nejistota \mathbf{w}_k . Energetické ohraničení je definováno jako

$$\mathbf{w}_0^F \in \mathcal{R} = \{\mathbf{x}_0^F \in \mathbb{R}^{(F+1)n_x} : (\mathbf{x}_0^F - \mathbf{w}_0^F)^T \mathbf{W} (\mathbf{x}_0^F - \mathbf{w}_0^F) \leq 1\} \quad (2.1)$$

a bodové ohraničení je definováno jako

$$\mathbf{w}_0^F \in \mathcal{R} = \{\mathbf{x}_0^F \in \mathbb{R}^{(F+1)n_x} : \|\mathbf{W}(\mathbf{x}_0^F - \mathbf{w}_0^F)\|_\infty \leq 1\}, \quad (2.2)$$

kde \mathbf{W} je známá matice. Návrh pomocného vstupního signálu za předpokladu energeticky ohraničených nejistot pro diskrétní i spojitě systémy je diskutován v [4], [61]–[64]. Formulace úlohy předpokládá dvojici lineárních modelů, jeden pro bezchybné chování a jeden pro chování systému v případě vzniku chyby, na konečném časovém horizontu. Energetické omezení se vztahuje na celkovou hodnotu energie stavového šumu, šumu měření a počáteční

podmínky stavu. Úkolem je navrhnout takový budicí vstupní signál, který dosáhne úplného oddělení množin vstupně-výstupních trajektorií, přičemž hodnoty vstupu jsou generovány v otevřené smyčce a následně aplikovány on-line. Odlišný pohled na neurčitosti je prezentován v [9], [11], [65], kde autoři předpokládají bodové omezení místo energetického omezení. Předpokládá se, že počáteční stav, stavové šумы a šумы měření jsou na konečném časovém horizontu ohraničeny zonotopy, neboli středově symetrickými konvexními polytopy. Vhodně navrženým pomocným vstupním signálem je možné pro lineární diskretní systémy zajistit, že měřený výstup všech možných chování systému leží uvnitř oddělených zonotopů. Metoda byla rozšířena pro omezení nejistot pomocí obecných polytopů [66]. Bodové omezení nejistot je dále uvažováno například v [10], [67].

2.3 Optimální aktivní detekce chyb a řízení

Jak bylo uvedeno v podkapitole 2.2, formulace úlohy aktivní detekce chyb využívá vícemodelového přístupu. V této části je představena pravděpodobnostní metoda aktivní detekce chyb založená na minimalizaci obecného návrhového kritéria. Výhodou této formulace je možnost volby návrhového kritéria, které zpravidla penalizuje chybná rozhodnutí detektoru chyb. V takovém případě se hovoří o aktivní detekci chyb. Navíc je možné kritérium rozšířit o požadavky na chování systému a tento případ pak představuje úlohu *aktivní detekce chyb a řízení*. Jestliže kritérium obsahuje pouze požadavky na řízení, formulovaná úloha představuje úlohu optimálního řízení.

Návrh optimálního budicího vstupního signálu a generátoru rozhodnutí na základě minimalizace obecného návrhového kritéria, které hodnotí kvalitu detekce chyb na konečném časovém horizontu, byl poprvé prezentován v [48]. Vývoj stavu a výstupu je modelován pomocí známých podmíněných hustot pravděpodobnosti a změna chování systému je modelována jako Markovův řetězec s konečným počtem stavů a známou maticí přechodů. Řešení formulované optimalizační úlohy je založeno na dynamickém programování [50] a Bellmanově rekurzivní rovnici. Metoda byla následně rozpracována pro lineární gaussovské systémy [68], nelineární systémy [69] a také pro lineární systémy za předpokladu omezujících podmínek [51], [70]. Následující část se zabývá pouze stručným představením formulace úlohy návrhu optimálního aktivního detektoru chyb a řízení spolu s jejím řešením pomocí dynamického programování. Bližší popis úlohy může být nalezen v [15], [71], [72].

2.3.1 Formulace úlohy

Model systému

Nechť je uvažován systém, jehož chování může být v každém časovém okamžiku $k \in \mathcal{T} = \{0, 1, \dots, F\}$ popsáno nelineárním diskretním stochastickým modelem

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{f}_k(\mathbf{x}_k, \mu_k, \mathbf{u}_k, \mathbf{w}_k), \\ \mu_{k+1} &= \mathbf{g}_k(\mathbf{x}_k, \mu_k, \mathbf{u}_k, \mathbf{e}_k), \\ \mathbf{y}_k &= \mathbf{h}_k(\mathbf{x}_k, \mu_k, \mathbf{v}_k), \end{aligned} \quad (2.3)$$

kde $\mathbf{x}_k \in \mathbb{R}^{n_x}$, $\mu_k \in \mathcal{M} = \{1, 2, \dots, N_\mu\}$, $\mathbf{u}_k \in \mathcal{U}_k \subset \mathbb{R}^{n_u}$, $\mathbf{y}_k \in \mathbb{R}^{n_y}$, $\mathbf{w}_k \in \mathbb{R}^{n_w}$, $\mathbf{v}_k \in \mathbb{R}^{n_v}$ a $\mathbf{e}_k \in \mathbb{R}^{n_e}$ jsou v pořadí spojitá část stavu, diskretní část stavu, vstup, výstup, stavový šum, šum měření a šum ovlivňující vývoj diskretní části stavu; $\mathbf{f}_k : \mathbb{R}^{n_x} \times \mathcal{M} \times \mathcal{U}_k \times \mathbb{R}^{n_w} \mapsto \mathbb{R}^{n_x}$, $\mathbf{h}_k : \mathbb{R}^{n_x} \times \mathcal{M} \times \mathbb{R}^{n_v} \mapsto \mathbb{R}^{n_y}$ a $\mathbf{g}_k : \mathbb{R}^{n_x} \times \mathcal{M} \times \mathcal{U}_k \times \mathbb{R}^{n_e} \mapsto \mathcal{M}$ jsou známé nelineární vektorové funkce spojitého stavu, měření a diskretního modelu. Spojitá část stavu \mathbf{x}_k spolu s diskretní částí stavu μ_k dohromady tvoří stav systému $\mathbf{s}_k = [\mathbf{x}_k^\top, \mu_k]^\top$. Spojitá část stavu \mathbf{x}_k obsahuje složky, které se vyvíjí spojitě, zatímco diskretní část stavu μ_k reprezentuje index modelu systému. Předpokládá se, že pro $\mu_k = 1$ je chování systému bezchybné, zatímco pro $\mu_k = i$, $i \in \{2, 3, \dots, N_\mu\}$ v systému nastala chyba. Diskretní část stavu μ_k tedy obsahuje informaci o chybě a umožňuje tak provést izolaci chyb. Dále se v úloze předpokládá, že je systém pozorovatelný, říditelný a počáteční stav \mathbf{s}_0 je popsán známou hustotou pravděpodobnosti $p(\mathbf{s}_0) = p(\mathbf{x}_0)p(\mu_0)$ ¹. Zároveň \mathbf{w}_k , \mathbf{v}_k a \mathbf{e}_k jsou vzájemně nezávislé bílé šumy nezávislé na počátečním hybridním stavu \mathbf{s}_0 , které jsou popsány známými hustotami pravděpodobností $p(\mathbf{w}_k)$, $p(\mathbf{v}_k)$ a $p(\mathbf{e}_k)$.

Aktivní detektor chyb a regulátor

Aktivní detektor chyb a regulátor představuje systém, který umožní na základě dostupné informace generovat v každém časovém okamžiku $k \in \mathcal{T}$ rozhodnutí $d_k \in \mathcal{M}$ o chybě v systému a vstup $\mathbf{u}_k \in \mathcal{U}_k$, který má dvojí funkci: regulovat vývoj spojitě části stavu \mathbf{x}_k a vybudit systém tak, aby se zvýšila kvalita budoucích rozhodnutí. Struktura aktivního detektoru chyb a regulátoru je následující

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \boldsymbol{\rho}_k(\mathbf{Z}_0^k) = \begin{bmatrix} \sigma_k(\mathbf{Z}_0^k) \\ \boldsymbol{\gamma}_k(\mathbf{Z}_0^k, d_k) \end{bmatrix}, \quad (2.4)$$

kde $\mathbf{Z}_0^k = [(\mathbf{y}_0^k)^\top, (\mathbf{u}_0^{k-1})^\top, (d_0^{k-1})^\top]^\top \in \mathcal{Z}_k$ je informační vektor, $\sigma_k : \mathcal{Z}_k \mapsto \mathcal{M}$ je neznámá funkce generátoru rozhodnutí, $\boldsymbol{\gamma}_k : \mathcal{Z}_k \times \mathcal{M} \mapsto \mathcal{U}_k$ je neznámá vektorová funkce vstupního signálu a společně tvoří neznámou vektorovou funkci $\boldsymbol{\rho}_k : \mathcal{Z}_k \mapsto \mathcal{M} \times \mathcal{U}_k$ z prostoru všech

¹Hybridní náhodnou veličinu \mathbf{s}_k lze popsat jako spojitou náhodnou veličinu za předpokladu popisu diskretní náhodné veličiny μ_k pomocí hustoty pravděpodobnosti $p(\mu_0) = \sum_{i=1}^{N_\mu} P(\mu_0)\delta(\mu_0 - i)$, kde δ je Diracova delta funkce, p značí hustotu pravděpodobnosti a P značí pravděpodobnost. Sdružená hustota pak může být zapsána ve tvaru $p(\mathbf{s}_0) = p(\mathbf{x}_0)p(\mu_0)$. Pro zjednodušení zápisu je tento popis uvažován ve zbytku práce.

přístupných funkcí. Množina \mathcal{Z}_k je zavedena z důvodu přehlednosti a je definována jako $\mathcal{Z}_k = \mathbb{R}^{(k+1) \times n_y} \times \mathcal{U}_0 \times \mathcal{U}_1 \times \dots \times \mathcal{U}_{k-1} \times \mathcal{M}^k$. Funkce generátoru rozhodnutí σ_k může být předem daná. Úkolem je pak najít funkci generátoru vstupního signálu γ_k , která respektuje daný generátor rozhodnutí. Druhá možnost nastane v případě předem stanovené funkce γ_k . Úkolem je navrhnout funkci σ_k vzhledem k známé funkci γ_k . Obě úlohy jsou blíže diskutovány například v [15], [71].

Návrhové kritérium

Cíle návrhu aktivního detektoru chyb a regulátoru mohou být vyjádřeny pomocí návrhového kritéria, které hodnotí očekávané ztráty na konečném časovém horizontu. K ohodnocení může být využita funkce se dvěma vlastnostmi:

- penalizací chybného rozhodnutí,
- ohodnocením trajektorie stavu a vstupu systému.

Pro danou úlohu může být definováno následující aditivní kritérium

$$J = \mathbb{E} \left\{ \sum_{k=0}^F L_k(\mu_k, \mathbf{x}_k, \mathbf{u}_k, d_k) \right\}, \quad (2.5)$$

kde $\mathbb{E}\{\cdot\}$ je operátor střední hodnoty přes všechny náhodné veličiny a $L_k : \mathcal{M} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathcal{M} \mapsto \mathbb{R}^+$ je nezáporná funkce vhodné struktury hodnotící kvalitu detekce chyb a řízení. Symbol \mathbb{R}^+ označuje množinu všech nezáporných reálných čísel $\mathbb{R}^+ = \langle 0, \infty \rangle$. Hodnotící funkce může mít následující strukturu, která umožňuje volit kompromis mezi detekcí chyb a řízením skrz váhový koeficient

$$L_k(\mu_k, \mathbf{x}_k, \mathbf{u}_k, d_k) = \alpha_k L_k^d(\mu_k, d_k) + (1 - \alpha_k) L_k^c(\mathbf{x}_k, \mathbf{u}_k), \quad (2.6)$$

kde $\alpha_k \in \langle 0, 1 \rangle$ je váhový koeficient, $L_k^d : \mathcal{M} \times \mathcal{M} \mapsto \mathbb{R}^+$ je ztrátová funkce reflektující požadavky na detekci chyb, $L_k^c : \mathbb{R}^{n_x} \times \mathcal{U}_k \mapsto \mathbb{R}^+$ je ztrátová funkce odrážející požadavky na trajektorie spojitě části stavu a vstupu. V závislosti na volbě váhového koeficientu α_k pro všechny časové okamžiky $k \in \mathcal{T}$ mohou nastat následující tři případy:

- úloha optimálního duálního řízení pro $\alpha_k = 0$,
- úloha optimální aktivní detekce chyb a řízení pro $\alpha_k \in (0, 1)$,
- úloha optimální aktivní detekce chyb pro $\alpha_k = 1$.

Zároveň je rozumné předpokládat, že ztrátová funkce detekce chyb splňuje následující nerovnost $L_k^d(\mu_k, \mu_k) \leq L_k^d(\mu_k, d_k)$ pro všechna $\mu_k \in \mathcal{M}$ a $d_k \in \mathcal{M}$.

Cílem návrhu optimálního detektoru chyb a regulátoru je najít takovou posloupnost funkcí $\boldsymbol{\rho}_0^F = [\boldsymbol{\rho}_0, \boldsymbol{\rho}_1, \dots, \boldsymbol{\rho}_F]$, která minimalizuje návrhové kritérium (2.5). Posloupnost funkcí $\boldsymbol{\rho}_0^F$ bývá často označována jako strategie a tento termín se bude používat i v této práci. Pro jednoduchost nechť je strategie označena jako $\boldsymbol{\rho} = \boldsymbol{\rho}_0^F$. Strategie minimalizující (2.5) se nazývá optimální strategie a je označena jako $\boldsymbol{\rho}^* = [\boldsymbol{\rho}_0^*, \boldsymbol{\rho}_1^*, \dots, \boldsymbol{\rho}_F^*]$.

2.3.2 Návrh řešení

Řešení optimalizační úlohy formulované v podkapitole 2.3.1 je založeno na využití Bellmanovy optimalizační rekurze, která spočívá v postupném vyčíslení minimální očekávané hodnoty koncové části kritéria optimality pro všechny možné hodnoty informačního vektoru. Optimální aktivní detektor a regulátor může být nalezen prostřednictvím řešení obecné Bellmanovy rekurzivní rovnice

$$V_k^*(\mathbf{Z}_0^k) = \min_{d_k \in \mathcal{M}, \mathbf{u}_k \in \mathcal{U}_k} \mathbb{E} \left\{ L_k(\mu_k, \mathbf{x}_k, \mathbf{u}_k, d_k) + V_{k+1}^*(\mathbf{Z}_0^{k+1}) | \mathbf{Z}_0^k, d_k, \mathbf{u}_k \right\}, \quad (2.7)$$

kde $k = F, F-1, \dots, 1, 0$ a $V_k^* : \mathcal{Z}_k \mapsto \mathbb{R}$ je optimální Bellmanova funkce definující minimální očekávanou hodnotu budoucích ztrát na základě dostupného informačního vektoru \mathbf{Z}_0^k . Počáteční podmínka zpětné rekurzivní rovnice (2.7) je $V_{F+1}^* = 0$. Na základě optimální Bellmanovy funkce lze určit optimální strategii $\boldsymbol{\rho}^*$. Postup řešení zpětné rekurzivní rovnice (2.7) je následující. V prvním kroku je určena optimální Bellmanova funkce V_F^* . Na základě V_F^* je určena optimální funkce $\boldsymbol{\rho}_F^*$, která se využije při dalším výpočtu optimální Bellmanovy funkce V_{F-1}^* . Tento postup je opakován, dokud není nalezena optimální strategie $\boldsymbol{\rho}^*$. Funkce popisující optimální rozhodnutí d_k^* a optimální vstup \mathbf{u}_k^* je v každém časovém okamžiku definována jako

$$\begin{bmatrix} d_k^* \\ \mathbf{u}_k^* \end{bmatrix} = \boldsymbol{\rho}_k^*(\mathbf{Z}_0^k) = \arg \min_{d_k \in \mathcal{M}, \mathbf{u}_k \in \mathcal{U}_k} \mathbb{E} \left\{ L_k(\mu_k, \mathbf{x}_k, \mathbf{u}_k, d_k) + V_{k+1}^*(\mathbf{Z}_0^{k+1}) | \mathbf{Z}_0^k, d_k, \mathbf{u}_k \right\}. \quad (2.8)$$

Minimální hodnotu kritéria (2.5) je možné vyjádřit využitím Bellmanovy funkce jako

$$J^* = J(\boldsymbol{\rho}^*) = \mathbb{E} \{ V_0^*(\mathbf{Z}_0) \}. \quad (2.9)$$

Při řešení úlohy je nutné v rovnicích (2.7) a (2.8) vyhodnotit podmíněné střední hodnoty. Z toho důvodu je zapotřebí pro každý časový okamžik $k \in \mathcal{T}$ nejdříve určit podmíněnou hustotu pravděpodobnosti trajektorie stavu $p(\mathbf{s}_0^k | \mathbf{Z}_0^k, d_k, \mathbf{u}_k)$ a prediktivní podmíněnou hustotu pravděpodobnosti měřeného výstupu $p(\mathbf{y}_{k+1} | \mathbf{Z}_0^k, d_k, \mathbf{u}_k)$. K nalezení těchto hustot pravděpodobnosti lze využít technik z oblasti nelineární filtrace [73].

Jedná se o obecnou úlohou optimální aktivní detekce chyb a řízení na konečném časovém horizontu, kde v závislosti na volbě hodnoticí funkce L a struktury systému může nastat devět speciálních případů aktivní detekce chyb a řízení. Jednotlivé případy jsou přiblíženy v [74], avšak důležitým případem je aktivní detekce chyb bez omezení, kterým se převážně zabývá právě tato práce.

Kapitola 3

Cíle disertační práce

Obsah této kapitoly je rozdělený do dvou navazujících částí. První část je věnována motivaci pro výzkum nových postupů v úloze detekce chyb v dynamických systémech. V druhé části jsou představeny dílčí cíle disertační práce. Cíle práce se zabývají rozšířením úlohy aktivní detekce chyb z konečného na nekonečný časový horizont. Nicméně dílčím cílem práce je zároveň rozvoj detekce změn a chyb bez znalosti modelu systému a možného propojení s úlohou optimálního řízení.

3.1 Motivace

Schopnost správného rozhodnutí roste se složitostí systémů. Správného a včasného odhalení chyb v systémech pomáhá zvyšovat bezpečnost, zachovat kvalitu a snižovat výdaje. Existují dva hlavní přístupy k detekci chyb v závislosti na interakci mezi systémem a detektorem [37], [38]. Pasivní detekce chyb představuje tradiční přístup, který je dobře vyvinutý a aplikovaný na různé reálné problémy. Tento přístup zahrnuje metody založené na redundanci senzorů, analýzu signálů v časové nebo frekvenční oblasti, nebo metody využívající model chování systému. Druhý a poměrně mladý přístup k detekci chyb, který je založený na interakci mezi detektorem a systémem za účelem zvýšení kvality detekce chyb, bývá označován jako aktivní přístup. Vhodně navrženým pomocným vstupním signálem s budicím charakterem je možné zkrátit čas detekce chyby nebo například snížit počet chybných rozhodnutí. V případě úlohy aktivní detekce chyb a řízení je možné navrhnout zákony řízení a detekce chyb společně.

Metody aktivní detekce chyb lze rozdělit do dvou skupin na deterministické a pravděpodobnostní [Š9], [64]. V případě deterministických metod se předpokládají omezené neurčitosti v systému, zatímco pravděpodobnostní metody uvažují, že neurčitosti v systému odpovídají náhodné veličině se známým pravděpodobnostním rozdělením. Zároveň je možné metody aktivní detekce chyb rozdělit podle uvažované délky horizontu na metody s konečným a metody s nekonečným časovým horizontem, přičemž většina deterministických [4], [11], [64], [67] a pravděpodobnostních [3], [14]–[16] metod uvedených v odborné literatuře je věnována úloze na konečném časovém horizontu.

V případě konečného časového horizontu je budicí vstupní signál navržen vždy na pevně daném časovém intervalu. Důvodem takového návrhu je snížení výpočetní náročnosti oproti nekonečnému horizontu a fakt, že navržený budicí signál může negativně ovlivnit kvalitu řízení systému. V praxi může být systém nepřetržitě monitorován pasivním detektorem a v kritických okamžicích lze využít aktivního detektoru chyb. Další možností je periodické monitorování systému aktivním detektorem chyb.

Analogií k návrhu budicího vstupního signálu aktivního detektoru chyb na nekonečném horizontu může být návrh optimálního řídicího signálu lineárního kvadratického regulátoru. Výsledkem návrhu pro nekonečný časový horizont je pouze jeden řídicí zákon, který zajišťuje nepřetržité řízení systému. Nevýhodou takového návrhu je vyšší teoretická složitost oproti konečnému časovému horizontu. To samé se dá říci i o návrhu aktivního detektoru chyb na nekonečném časovém horizontu.

Předchozí úvahy jsou postavené na předpokladu znalosti modelu chování systému. Avšak v řadě úloh není model systému známý a jeho získání vyžaduje značné úsilí. Zadání úlohy proto může spočívat v detekci změn a chyb v systému pouze na základě vstupně-výstupních dat. Jedná se tedy o detekci chyb založenou na datech. K řešení takové úlohy lze využít některé postupy z umělé inteligence [28], [75].

3.2 Cíle

Na základě předchozí motivace jsou stanoveny cíle disertační práce, které jsou rozděleny do dvou oblastí. Hlavní oblastí výzkumu je optimální aktivní detekce chyb na nekonečném časovém horizontu. Druhou oblastí výzkumu je úloha detekce chyb bez znalosti modelu systému. Jelikož se jednotlivé cíle disertační práce v kapitolách prolínají, je pro přehlednost řešení dílčích cílů v jednotlivých kapitolách ilustrováno na obrázku 3.1.

3.2.1 Optimální aktivní detekce chyb na nekonečném časovém horizontu

V současné době existuje stále pouze omezené množství metod, které se zaměřují na návrh aktivního detektoru chyb na nekonečném časovém horizontu. První pokusy o rozšíření metody aktivní detekce chyb z podkapitoly 2.3 se týkaly návrhu aktivního detektoru chyb pro Markovovy řetězce [76] a pro lineární gaussovské systémy [77]. Nicméně obecnější formulace úlohy a návrh řešení pro nekonečný časový horizont nebyla doposud v literatuře diskutována.

Cíl 1: Základní cíl této práce spočívá ve formulaci úlohy návrhu aktivního detektoru chyb na nekonečném časovém horizontu. Pozornost je zaměřena na představení obecné formulace úlohy pro diskrétní stochastické dynamické systémy.

Cíl 2: Další krok spočívá v návrhu řešení formulované úlohy.

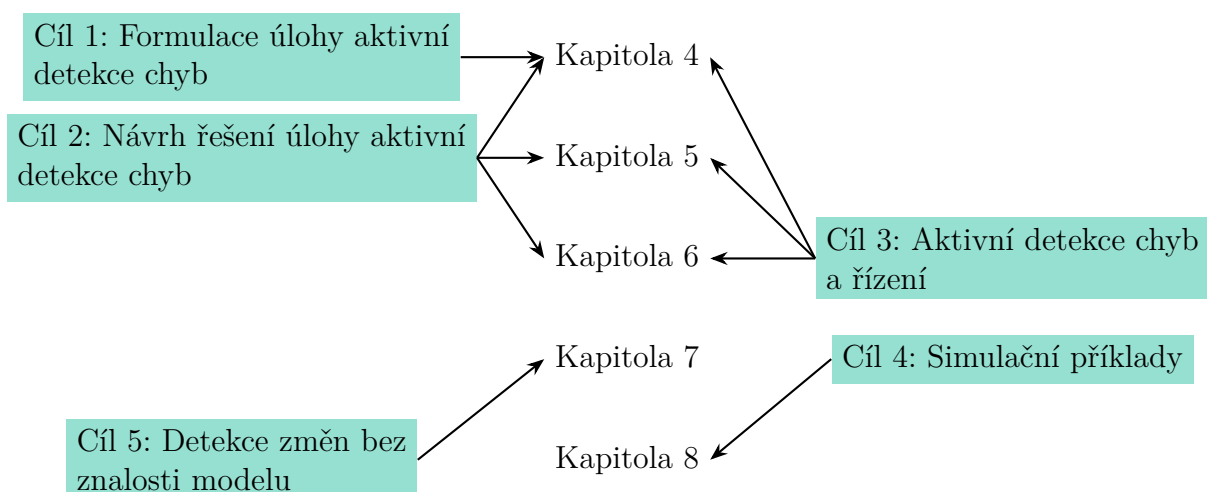
Cíl 3: Dílčím cílem práce je rozšíření formulace a řešení úlohy aktivní detekce chyb o požadavky na chování systému.

Cíl 4: Návrh aktivního detektoru chyb a jeho aplikace na reálný problém není triviální. Přesto řada autorů dokázala představit vlastnosti své metody v numerický případech [11], [14], [78]. Z těchto důvodů je dílčím cílem disertační práce ilustrovat vlastnosti vybraných suboptimálních řešení na numerických příkladech.

3.2.2 Detekce změn a chyb bez znalosti modelu

Přestože je hlavním předmětem zájmu této práce aktivní detekce chyb, motivující úlohou potřebnou při návrhu optimálního řízení bez znalosti modelu [79] je právě detekce změny chování na základě dostupných měření, kdy je zapotřebí určit vztah mezi nasbíranými daty a změnami chování systému.

Cíl 5: Cílem disertační práce je navrhnout a ověřit postup detekce změny chování na základě měřených vstupně-výstupních dat bez znalosti modelu systému.



Obrázek 3.1: Ilustrace řešení dílčích cílů disertační práce v jednotlivých kapitolách.

Kapitola 4

Formulace úlohy aktivní detekce chyb na nekonečném časovém horizontu

V této kapitole je formulována úloha návrhu optimálního aktivního detektoru chyb pro stochastické diskrétní systémy a obecné návrhové kritérium na nekonečném časovém horizontu. Její řešení bude diskutováno v kapitolách 5 a 6. Model systému, struktura aktivního detektoru chyb a návrhové kritérium jsou popsány v podkapitole 4.1. Předpokládá se, že chování systému může být modelováno konečnou množinou diskrétních stochastických stavových modelů a přechody mezi modely se řídí známou maticí pravděpodobností přechodů. Cílem úlohy je určit takový zákon generující budící vstupy a rozhodnutí, který minimalizuje ztráty způsobené chybou detekcí, tedy nesouladem rozhodnutí a aktuálního modelu. Podkapitola 4.2 se zabývá popisem modelu s neúplnou informací o stavu. Závěrečná podkapitola 4.3 se zabývá formulací úlohy návrhu aktivního detektoru chyb a regulátoru.

4.1 Formulace úlohy

4.1.1 Model systému

Postup návrhu aktivního detektoru chyb předpokládá, že chování systému může být popsáno množinou známých stavových modelů, které se mezi sebou v čase přepínají v závislosti na známých pravděpodobnostech přechodů mezi jednotlivými modely. Stav systému se v tomto případě skládá ze dvou komponent: spojité části představující běžný stav a diskrétní části, která přímo definuje model systému. Obecně lze předpokládat, že měřeným výstupem systému není přímo stav, ale nějaká funkce stavu. Zároveň je důležité při modelování systému zohlednit různé chyby modelování nebo poruchy působící na stav a výstup systému. Formulace úlohy uvažuje systémy, které jsou říditelné a pozorovatelné.

Nechť je chování systému v každém časovém okamžiku $k \in \mathcal{T} = \{0, 1, 2, \dots\}$ popsáno následujícím diskrétním stochastickým stavovým modelem

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mu_k, \mathbf{u}_k) + \mathbf{G}_{\mu_k} \mathbf{w}_k, \\ \mathbf{y}_k &= \mathbf{g}(\mathbf{x}_k, \mu_k) + \mathbf{H}_{\mu_k} \mathbf{v}_k,\end{aligned}\tag{4.1}$$

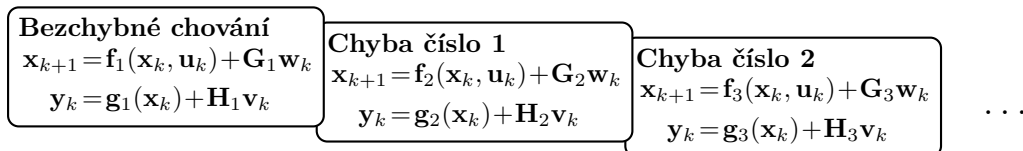
kde $\mathbf{x}_k \in \mathcal{X} = \mathbb{R}^{n_x}$ je spojitá část stavu, $\mu_k \in \mathcal{M} = \{1, 2, \dots, N_\mu\}$ je diskrétní část stavu označovaná jako index modelu, $\mathbf{u}_k \in \mathcal{U} \subset \mathbb{R}^{n_u}$ je vstup ze známé konečné množiny možných vstupů $\mathcal{U} = \{\bar{\mathbf{u}}^1, \bar{\mathbf{u}}^2, \dots, \bar{\mathbf{u}}^{N_u}\}$, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ je měřený výstup, $\mathbf{w}_k \in \mathbb{R}^{n_x}$ je stavový šum a $\mathbf{v}_k \in \mathbb{R}^{n_y}$ je šum měření. Funkce $\mathbf{f} : \mathcal{X} \times \mathcal{M} \times \mathcal{U} \mapsto \mathcal{X}$ spolu s $\mathbf{g} : \mathcal{X} \times \mathcal{M} \mapsto \mathbb{R}^{n_y}$ jsou známé vektorové funkce, které mají následující strukturu

$$\mathbf{f}(\mathbf{x}_k, \mu_k, \mathbf{u}_k) = \begin{cases} \mathbf{f}_1(\mathbf{x}_k, \mathbf{u}_k) & \text{pro } \mu_k = 1, \\ \mathbf{f}_2(\mathbf{x}_k, \mathbf{u}_k) & \text{pro } \mu_k = 2, \\ \vdots & \\ \mathbf{f}_{N_\mu}(\mathbf{x}_k, \mathbf{u}_k) & \text{pro } \mu_k = N_\mu, \end{cases} \quad \mathbf{g}(\mathbf{x}_k, \mu_k) = \begin{cases} \mathbf{g}_1(\mathbf{x}_k) & \text{pro } \mu_k = 1, \\ \mathbf{g}_2(\mathbf{x}_k) & \text{pro } \mu_k = 2, \\ \vdots & \\ \mathbf{g}_{N_\mu}(\mathbf{x}_k) & \text{pro } \mu_k = N_\mu, \end{cases} \quad (4.2)$$

kde $\mathbf{f}_i : \mathcal{X} \times \mathcal{U} \mapsto \mathcal{X}$ a $\mathbf{g}_i : \mathcal{X} \mapsto \mathbb{R}^{n_y}$ jsou známé funkce popisující i -tý model. Předpokládá se, že $\mathbf{G}_{\mu_k} \in \mathbb{R}^{n_x \times n_x}$ spolu s $\mathbf{H}_{\mu_k} \in \mathbb{R}^{n_y \times n_y}$ jsou regulární matice příslušných šumů známé pro všechny indexy modelů a \mathbf{w}_k spolu s \mathbf{v}_k jsou vzájemně nezávislé bílé šумы nezávislé na počátečním stavu popsané příslušnými známými hustotami pravděpodobnosti p_{w_k} a p_{v_k} s nulovými středními hodnotami a jednotkovými kovariančními maticemi. Trajektorie indexu modelu se řídí Markovovým procesem se známými pravděpodobnostmi přechodu

$$P_{j,i} = P(\mu_{k+1} = j | \mu_k = i), \quad (4.3)$$

pro všechny $i, j \in \mathcal{M}$, které tvoří matici pravděpodobností přechodů $\mathbf{P} \in \mathbb{R}^{N_\mu \times N_\mu}$. Počáteční index modelu μ_0 je popsán známou pravděpodobností $P(\mu_0)$ a počáteční spojitá část stavu \mathbf{x}_0 je popsána známým rozdělením pravděpodobnosti p_{x_0} .



Obrázek 4.1: Ilustrace modelů pro jednotlivá chování systému.

Struktura modelu (4.1) je velmi obecná a umožňuje široké použití. Jak je zobrazeno na obrázku 4.1, bezchybné chování systému je popsáno stavovým modelem pro index modelu $\mu_k = 1$, zatímco chybná chování systému odpovídají indexům modelu $\mu_k = i, i \in \{2, 3, \dots, N_\mu\}$. Spojitá část stavu \mathbf{x}_k spolu s indexem modelu μ_k tvoří dohromady neznámý stav systému $\mathbf{s}_k = [\mathbf{x}_k^T, \mu_k]^T \in \mathcal{S} = \mathcal{X} \times \mathcal{M}$. Z hlediska pojetí úlohy mohou nastat dva speciální případy v závislosti na interpretaci měřeného výstupu systému \mathbf{y}_k .

Úloha s nepřímo měřitelnou spojitou částí stavu je obecná úloha popsána modelem (4.1), kde spojitá část stavu \mathbf{x}_k je měřena nepřímo jako $\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mu_k) + \mathbf{H}_{\mu_k} \mathbf{v}_k$.

Úloha s přímo měřitelnou spojitou částí stavu představuje případ, kdy spojitá část stavu \mathbf{x}_k je měřena přímo, neboli $\mathbf{y}_k = \mathbf{x}_k$. Chování systému je následně popsáno stavovým modelem

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mu_k, \mathbf{u}_k) + \mathbf{G}_{\mu_k} \mathbf{w}_k \quad (4.4)$$

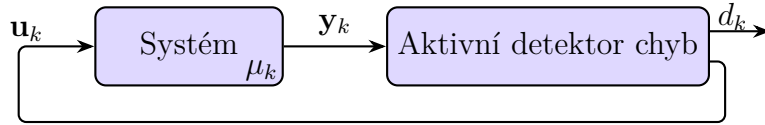
společně s pravděpodobnostmi přechodů (4.3).

4.1.2 Aktivní detektor chyb

Aktivní detektor chyb může být definován jako funkce, která zobrazuje veškerou dostupnou informaci na rozhodnutí o chybě v systému a vstupní budící signál. Úkolem vstupního signálu je vybudit systém, a tím zlepšit kvalitu detekce. Aktivní detektor chyb lze v každém časovém okamžiku $k \in \mathcal{T}$ definovat jako

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \boldsymbol{\rho}_k(\mathbf{Z}_0^k) = \begin{bmatrix} \sigma_k(\mathbf{Z}_0^k) \\ \boldsymbol{\gamma}_k(\mathbf{Z}_0^k) \end{bmatrix}, \quad (4.5)$$

kde $\mathbf{Z}_0^k = [(\mathbf{y}_0^k)^\top, (\mathbf{u}_0^{k-1})^\top] \in \mathcal{Z}_k$ je známý informační vektor zavedený z důvodu zjednodušení zápisu, $\boldsymbol{\rho}_k : \mathcal{Z}_k \mapsto \mathcal{M} \times \mathcal{U}$ je neznámá nelineární vektorová funkce vektorového argumentu z množiny všech přípustných funkcí představující aktivní detektor chyb, $\sigma_k : \mathcal{Z}_k \mapsto \mathcal{M}$ je funkce generátoru rozhodnutí a $\boldsymbol{\gamma}_k : \mathcal{Z}_k \mapsto \mathcal{U}$ je vektorová funkce generátoru vstupního signálu. Základní blokové schéma systému a aktivního detektoru chyb je ilustrováno na obrázku 4.2. Z obrázku je patrné, že aktivní detektor chyb generuje rozhodnutí o chybě d_k a budící vstup \mathbf{u}_k , který vstupuje zpět do systému. K určení zákona generujícího d_k a \mathbf{u}_k je nutné stanovení návrhových požadavků.



Obrázek 4.2: Blokový diagram zapojení systému a aktivního detektoru chyb.

V případě úlohy s přímo měřitelnou spojitou částí stavu je aktivní detektor chyb definován jako

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \boldsymbol{\rho}_k(\mathbf{Y}_0^k) = \begin{bmatrix} \sigma_k(\mathbf{Y}_0^k) \\ \boldsymbol{\gamma}_k(\mathbf{Y}_0^k) \end{bmatrix}, \quad (4.6)$$

kde $\mathbf{Y}_0^k = [(\mathbf{x}_0^k)^\top, (\mathbf{u}_0^{k-1})^\top]^\top \in \mathcal{D}_k$ je informační vektor. Na rozdíl od systémů s nepřímou měřitelnou spojitou částí stavu je tak současné rozhodnutí o chybě d_k a vstup \mathbf{u}_k generován na základě měřených spojitých částí stavů a vstupů.

4.1.3 Návrhové kritérium

Aktivní detektor chyb je zapotřebí navrhnout tak, aby vyhovoval návrhovým požadavkům. Měřítkem kvality aktivního detektoru chyb na nekonečném časovém horizontu pro formulovanou úlohu může být následující diskontované aditivní kritérium

$$J = \lim_{F \rightarrow \infty} \mathbb{E} \left\{ \sum_{k=0}^F \eta^k L^d(\mu_k, d_k) \right\}, \quad (4.7)$$

kde $L^d : \mathcal{M} \times \mathcal{M} \mapsto \mathbb{R}^+$ je obecná ztrátová funkce penalizující nesprávná rozhodnutí, neboli nesoulad mezi skutečným indexem modelu μ_k a rozhodnutím d_k , $\mathbb{R}^+ = \langle 0, \infty \rangle$ a $\eta \in (0, 1)$ je diskontní faktor, který snižuje význam budoucích ztrát s rostoucím časem. Předpokládá se, že t-invariantní funkce L^d je vhodně zvolena tak, aby byla splněna následující podmínka

$$L_{\min}^d \leq L^d(\mu_k, d_k) \leq L_{\max}^d, \quad (4.8)$$

kde $L_{\min}^d = \min_{\mu_k, d_k} L^d(\mu_k, d_k) \geq 0$ je dolní mez a $L_{\max}^d = \max_{\mu_k, d_k} L^d(\mu_k, d_k) < \infty$ je horní mez penalizace kvality detekce chyb. Podmínka (4.8) stanovuje omezení na minimální a maximální hodnotu ztrátové funkce. Tím je zajištěno, že hodnota kritéria (4.7) je konečná. Aby úloha měla smysl z hlediska detekce chyb, ztrátová funkce navíc musí penalizovat správná rozhodnutí méně než chybná rozhodnutí, neboli

$$L^d(\mu_k, \mu_k) < L^d(\mu_k, d_k), \quad d_k \in \mathcal{M}, d_k \neq \mu_k. \quad (4.9)$$

Cílem návrhu aktivního detektoru chyb je najít takovou optimální strategii $\boldsymbol{\rho}^* = \{\boldsymbol{\rho}_k^*\}_{k=0}^\infty$, která minimalizuje kritérium (4.7), neboli

$$\boldsymbol{\rho}^* = \arg \min_{\boldsymbol{\rho} \in \mathcal{A}} J, \quad (4.10)$$

kde \mathcal{A} je množina všech přípustných strategií.

4.2 Úloha s neúplnou informací o stavu

Index modelu μ_k spolu se spojitou částí stavu \mathbf{x}_k tvoří dohromady stav systému \mathbf{s}_k . Jelikož index modelu ani spojitá část stavu nejsou známé, formulovaná optimalizační úloha může být označena jako úloha s neúplnou informací o stavu. Řešení těchto úloh je složitější v porovnání s řešením úloh s úplnou informací o stavu, kdy stav systému je přímo měřitelný [50]. Kromě teoretické složitosti je zapotřebí použít výpočetně náročnější algoritmy k nalezení aproximativního řešení. Z těchto důvodů se zdá být výhodné transformovat původně formulovanou úlohu na úlohu s úplnou informací o stavu. V případě řešení optimalizačních úloh s úplnou informací o stavu je možné využít některý ze známých optimalizačních nástrojů v závislosti na formulaci úlohy. Mezi běžně používané optimalizační nástroje patří lineární programování [80], nelineární programování [81], celočíselné a smíšené celočíselné programování [82], stochastické programování [83], nebo například dynamické programování [50], [84].

4.2.1 Postačující statistiky

Úkolem této části je popsat nahrazení neznámého stavu \mathbf{s}_k tak, aby formulovaná úloha z podkapitoly 4.1 mohla být transformována na úlohu s úplnou informací o stavu. Zhruba řečeno, v pravděpodobnosti a statistice se využívají tzv. *postačující statistiky* k nahrazení

náhodného výběru podmíněného neznámým parametrem tak, aby rozdělení náhodného výběru podmíněné postačujícími statistikami bylo nezávislé na daném neznámém parametru [85]. Postačující statistiky mohou být využity i v popisované úloze. Cílem je tedy najít takovou statistiku, která nahradí informační vektor \mathbf{Z}_0^k a bude obsahovat stejnou informaci o μ_k a \mathbf{x}_k .

Veškerá informace o současném stavu \mathbf{s}_k za předpokladu znalosti informačního vektoru \mathbf{Z}_0^k je s využitím Bayesova přístupu k rekurzivní filtraci obsažena v podmíněné hustotě pravděpodobnosti $p(\mathbf{s}_k|\mathbf{Z}_0^k)$, která může sloužit jako postačující statistika stavu \mathbf{s}_k . Výpočet podmíněné hustoty pravděpodobnosti $p(\mathbf{s}_k|\mathbf{Z}_0^k)$ je úlohou odhadu stavu, jejíž řešení je obecně dáno následujícími vztahy

$$p(\mathbf{s}_k|\mathbf{Z}_0^k) = \frac{p(\mathbf{y}_k|\mathbf{s}_k)p(\mathbf{s}_k|\mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1})}{\int \sum_{\mu_k} p(\mathbf{y}_k|\mathbf{s}_k)p(\mathbf{s}_k|\mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1})d\mathbf{x}_k}, \quad (4.11)$$

$$p(\mathbf{s}_{k+1}|\mathbf{Z}_0^k, \mathbf{u}_k) = \int \sum_{\mu_k} p(\mathbf{s}_{k+1}|\mathbf{s}_k, \mathbf{u}_k)p(\mathbf{s}_k|\mathbf{Z}_0^k)d\mathbf{x}_k, \quad (4.12)$$

kde $p(\mathbf{s}_0|\mathbf{Z}_0^{-1}, \mathbf{u}_{-1}) = p(\mathbf{s}_0)$ je počáteční podmínka. Vztah (4.11) je Bayesův vztah, (4.12) je Chapman-Kolmogorovova rovnice a společně jsou vztahy (4.11) a (4.12) v této práci označeny jako Bayesovy rekurzivní vztahy (BRV). Obecné řešení úlohy odhadu stavu je dáno filtračním krokem (4.11) a jednokrokovou predikcí (4.12). Jmenovatel (4.11) tvoří normalizační konstanta popisující vliv měřeného výstupu \mathbf{y}_k a $p(\mathbf{s}_{k+1}|\mathbf{Z}_0^k, \mathbf{u}_k)$ je prediktivní pravděpodobnost stavu. Rekurzivní vztahy (4.11) a (4.12) popisují vývoj podmíněné hustoty pravděpodobnosti stavu v čase.

K odhadu stavu \mathbf{s}_k je užitečné detailněji vysvětlit strukturu podmíněné hustoty pravděpodobnosti $p(\mathbf{s}_k|\mathbf{Z}_0^k)$. Hustota pravděpodobnosti stavu může být také zapsána jako

$$p(\mathbf{s}_k|\mathbf{Z}_0^k) = p(\mathbf{x}_k, \mu_k|\mathbf{Z}_0^k) = \sum_{\mu_0^{k-1}} p(\mathbf{x}_k|\mathbf{Z}_0^k, \mu_0^k)P(\mu_0^k|\mathbf{Z}_0^k), \quad (4.13)$$

kde $P(\mu_0^k|\mathbf{Z}_0^k)$ je podmíněná pravděpodobnost posloupnosti modelů, $p(\mathbf{x}_k|\mathbf{Z}_0^k, \mu_0^k)$ je podmíněná hustota pravděpodobnosti \mathbf{x}_k a $\sum_{\mu_0^{k-1}}$ značí součet přes všechny možné posloupnosti modelů μ_0^{k-1} . K získání $p(\mathbf{s}_k|\mathbf{Z}_0^k)$ je tedy zapotřebí objasnit postup výpočtu $P(\mu_0^k|\mathbf{Z}_0^k)$ a $p(\mathbf{x}_k|\mathbf{Z}_0^k, \mu_0^k)$.

Pravděpodobnost $P(\mu_0^k|\mathbf{Z}_0^k)$ může být vyjádřena pomocí BRV jako

$$P(\mu_0^k|\mathbf{Z}_0^k) = \frac{p(\mathbf{y}_k|\mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1}, \mu_0^k)P(\mu_0^k|\mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1})}{p(\mathbf{y}_k|\mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1})}, \quad (4.14)$$

kde podmíněná hustota pravděpodobnosti $p(\mathbf{y}_k|\mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1})$ je definována jako

$$p(\mathbf{y}_k|\mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1}) = \sum_{\mu_0^k} p(\mathbf{y}_k|\mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1}, \mu_0^k)P(\mu_0^k|\mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1}). \quad (4.15)$$

Diskrétní hustota pravděpodobnosti $P(\mu_0^k | \mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1})$ určuje rozložení pravděpodobnosti posloupnosti indexů modelu μ_0^k na základě pravděpodobností $P(\mu_0^{k-1} | \mathbf{Z}_0^{k-1})$ a pravděpodobnosti přechodů mezi modely $P(\mu_k | \mu_{k-1})$. Jelikož je index modelu μ_k nezávislý vstupu \mathbf{u}_{k-1} , může být podmíněná hustota pravděpodobnosti $P(\mu_0^k | \mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1})$ vyjádřena jako

$$P(\mu_0^k | \mathbf{Z}_0^{k-1}) = \begin{cases} P(\mu_0) & \text{pokud } k=0, \\ P(\mu_k | \mu_{k-1}) P(\mu_0^{k-1} | \mathbf{Z}_0^{k-1}) & \text{pokud } k > 0, \end{cases} \quad (4.16)$$

kde $P(\mu_k | \mu_{k-1})$ lze přímo získat z matice pravděpodobnosti přechodu \mathbf{P} .

Pokud by byl použit filtrační algoritmus poskytující $p(\mathbf{s}_k | \mathbf{Z}_0^k)$, může být vytvořen nový model systému, který pro úplnou informaci o stavu systému používá nekonečně rozměrný stav reprezentovaný právě podmíněnou hustotou pravděpodobnosti $p(\mathbf{s}_k | \mathbf{Z}_0^k)$. Nový model systému může být definován jako

$$p(\mathbf{s}_{k+1} | \mathbf{Z}_0^{k+1}) = \varphi(p(\mathbf{s}_k | \mathbf{Z}_0^k), \mathbf{u}_k, \mathbf{y}_{k+1}), \quad (4.17)$$

kde $\varphi : \mathcal{L} \times \mathcal{U} \times \mathbb{R}^{n_y} \mapsto \mathcal{L}$ je zobrazení popisující transformaci hustoty pravděpodobnosti $p(\mathbf{s}_k | \mathbf{Z}_0^k)$ v současném časovém okamžiku k na hustotu pravděpodobnosti $p(\mathbf{s}_{k+1} | \mathbf{Z}_0^{k+1})$ v budoucím časovém okamžiku $k+1$ v závislosti na současném známém vstupu \mathbf{u}_k a budoucím neznámém výstupu \mathbf{y}_{k+1} , který může být získán na základě hustoty pravděpodobnosti $p_{y_{k+1}}$. Prostor všech takových možných hustot pravděpodobnosti je označen symbolem \mathcal{L} . Vztah (4.17) nahrazuje model (4.1) a (4.3).

4.3 Aktivní detekce chyb a řízení

Formulace uvedená v podkapitole 4.1 se zabývá úlohou optimální aktivní detekce chyb bez požadavků na chování systému. Rozšíření takové úlohy na úlohu aktivní detekce chyb a řízení je poměrně snadné. Důležitou roli hraje ztrátová funkce, která musí hodnotit jak kvalitu detekce chyb, tak kvalitu řízení. Z hlediska struktury se v případě aktivní detekce chyb se hovoří o generátoru rozhodnutí a generátoru vstupního signálu. Vstupní signál má v takovém případě pouze budicí charakter za účelem zvýšení kvality detekce chyb. V úloze aktivní detekce chyb a řízení je předmětem zájmu návrh generátoru rozhodnutí a regulátoru. Ten budí systém za účelem zvýšení kvality detekce chyb a zároveň řídí trajektorii spojitě části stavu systému. Detektor chyb a regulátor jsou navrženy dohromady oproti úloze pasivní detekce chyb a řízení, kdy regulátor bývá navržen odděleně od detektoru chyb.

Nechť je chování systému popsáno modelem (4.1) a nechť je aktivní detektor chyb a regulátor popsán podobně jako ve (4.5) s tím rozdílem, že strategie generující řídicí vstup systému a samotný aktivní detektor chyb a regulátor jsou pro přehlednost označena horním indexem "r", neboli

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \boldsymbol{\rho}_k^r(\mathbf{Z}_0^k) = \begin{bmatrix} \sigma_k(\mathbf{Z}_0^k) \\ \gamma_k^r(\mathbf{Z}_0^k) \end{bmatrix}, \quad (4.18)$$

kde $\boldsymbol{\rho}_k^r : \mathcal{Z}_k \mapsto \mathcal{M} \times \mathcal{U}$, $\sigma_k : \mathcal{Z}_k \mapsto \mathcal{M}$ a $\boldsymbol{\gamma}_k^r : \mathcal{Z}_k \mapsto \mathcal{U}$ jsou v pořadí aktivní detektor chyb a regulátor, generátor rozhodnutí a generátor řídicího signálu. Cílem návrhu je najít takovou strategii $\boldsymbol{\rho}^r = \{\boldsymbol{\rho}_k^r\}_{k=0}^\infty$, která minimalizuje kritérium hodnotící kvalitu detekce chyb a cíle řízení. Struktura návrhového kritéria je zvolena stejně jako ve (4.7), neboli aditivní diskontované kritérium hodnotící ztráty na nekonečném časovém horizontu

$$J = \lim_{F \rightarrow \infty} \mathbb{E} \left\{ \sum_{k=0}^F \eta^k L(\mathbf{x}_k, \mu_k, d_k, \mathbf{u}_k) \right\}, \quad (4.19)$$

kde $L : \mathbb{R}^{n_x} \times \mathcal{M} \times \mathcal{M} \times \mathcal{U} \mapsto \mathbb{R}^+$ je ztrátová funkce. Klíčový rozdíl spočívá v definici ztrátové funkce, která v případě aktivní detekce chyb a řízení hodnotí kompromis mezi kvalitou detekce chyb a řízením. Taková ztrátová funkce může být definována jako

$$L(\mathbf{x}_k, \mu_k, d_k, \mathbf{u}_k) = \alpha L^d(\mu_k, d_k) + (1 - \alpha) L^c(\mathbf{x}_k, \mathbf{u}_k), \quad (4.20)$$

kde $\alpha \in \langle 0, 1 \rangle$ je váhový koeficient určující kompromis mezi požadavky na detekci chyb a požadavky na řízení systému, $L^d : \mathcal{M} \times \mathcal{M} \mapsto \mathbb{R}^+$ je ztrátová funkce hodnotící kvalitu detekce a $L^c : \mathbb{R}^{n_x} \times \mathcal{U} \mapsto \mathbb{R}^+$ je ztrátová funkce hodnotící kvalitu řízení. V závislosti na volbě váhového koeficientu α mohou nastat tři případy:

- úloha optimálního řízení pro $\alpha = 0$,
- úloha aktivní detekce chyb a řízení pro $\alpha \in (0, 1)$,
- úloha aktivní detekce chyb pro $\alpha = 1$.

Úloha aktivní detekce chyb ($\alpha = 1$) byla diskutována v podkapitole 4.1. Zároveň je nutné připomenout, že funkce L^d hodnotící kvalitu detekce musí splňovat podmínky (4.8) a (4.9) a funkce L^c hodnotící kvalitu řízení musí splňovat následující nerovnost

$$L_{\min}^c \leq L^c(\mathbf{x}_k, \mathbf{u}_k) \leq L_{\max}^c, \quad (4.21)$$

kde $L_{\min}^c = \sup_{\mathbf{x}_k, \mathbf{u}_k} L^c(\mathbf{x}_k, \mathbf{u}_k) \geq 0$ je dolní mez a $L_{\max}^c = \inf_{\mathbf{x}_k, \mathbf{u}_k} L^c(\mathbf{x}_k, \mathbf{u}_k) < \infty$ je horní mez penalizace kvality řízení. Uvedenými podmínkami je zaručeno, že hodnota kritéria (4.19) je konečná.

Kapitola 5

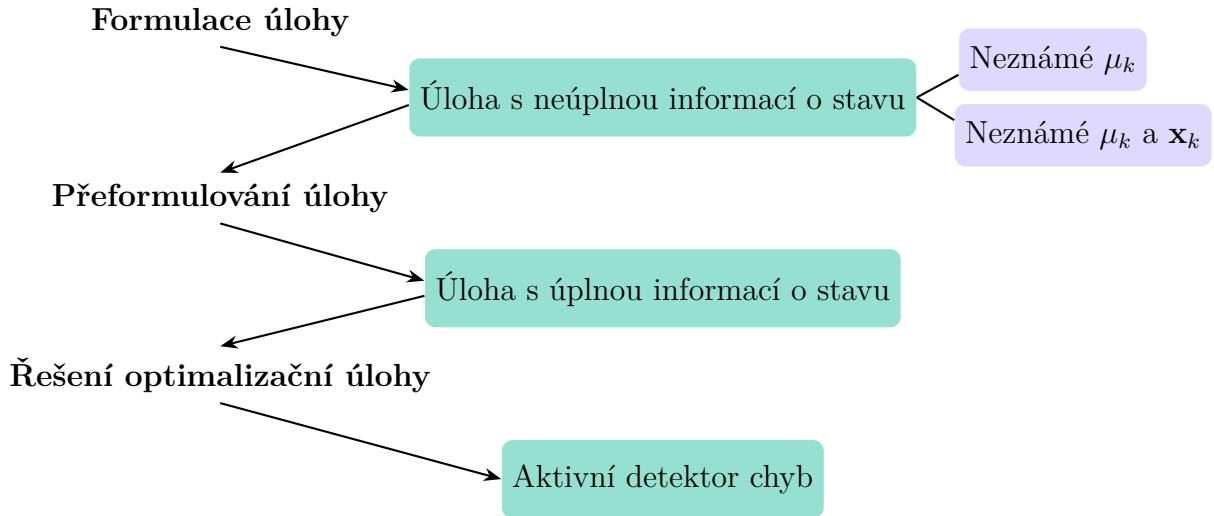
Přeformulování úlohy

Cíl této části spočívá v představení postupů, které umožňují převést původní optimalizační úlohu návrhu aktivního detektoru chyb formulovanou v kapitole 4 na ekvivalentní úlohu. Smyslem přeformulování je zjednodušení řešení optimalizační úlohy. Celý postup návrhu aktivního detektoru chyb je ilustrován na obrázku 5.1. Původně formulovanou úlohu s neúplnou informací o stavu lze přeformulovat jako úlohu s úplnou informací o stavu pomocí postačujících statistik stavu \mathbf{s}_k . Pro implementaci řešení však nejsou postačující statistiky příliš vhodné z důvodu jejich rozměrnosti. Postačujících statistik může být nekonečně mnoho, a proto je v této kapitole diskutována možnost aproximace postačujících statistik pomocí *informačních statistik* [0]. Hustota pravděpodobnosti náhodné veličiny lze ve speciálních případech nahradit konečným počtem veličin. Příkladem je nahrazení gaussovské hustoty pravděpodobnosti pomocí prvních dvou momentů náhodné veličiny. Možné formy informačních statistik pro různé modely systémů jsou uvedeny v podkapitole 5.1. Postup samotného přeformulování úlohy je navržen v podkapitole 5.2. Řešením optimalizační úlohy s úplnou informací o stavu se zabývá následující kapitola 6.

5.1 Informační statistiky

Jak bylo uvedeno v podkapitole 4.2.1, veškerá informace o stavu $\mathbf{s}_k = [\mathbf{x}_k^T, \mu_k]^T$ obsažená v informačním vektoru \mathbf{Z}_0^k může být reprezentována nekonečně rozměrnou statistikou v podobě podmíněné hustoty pravděpodobnosti $p(\mathbf{s}_k | \mathbf{Z}_0^k)$. Tato postačující statistika však z důvodu výpočetních nároků není příliš vhodná, proto je zapotřebí najít konečné statistiky, které hustotu pravděpodobnosti $p(\mathbf{s}_k | \mathbf{Z}_0^k)$ mohou nahradit. Takové statistiky jsou v této práci nazývány informačními statistikami. Pomocí informačních statistik lze původní úlohu s neúplnou informací o stavu přeformulovat na úlohu s úplnou informací o stavu.

Ještě před uvedením konkrétních informačních statistik a způsobu jejich získání je vhodné objasnit samotné přeformulování úlohy. Grafické znázornění přeformulování úlohy aktivní detekce chyb je zobrazeno na obrázku 5.2. V prvním kroku je na základě struktury modelu systému získán vektor informačních statistik $\boldsymbol{\xi}_k$. Na základě známých informačních statistik může být model systému, kritérium a aktivní detektor chyb transformován na opti-



Obrázek 5.1: Ilustrace postupu řešení optimalizační úlohy aktivní detekce chyb s obecným návrhovým kritériem kvality na nekonečném časovém horizontu.

malizační úlohu s úplnou informací o stavu, která bude reflektovat původní formulaci úlohy. Nový model systému je definován funkcí ϕ , kritérium obsahuje novou ztrátovou funkci \bar{L}^d a cílem je najít novou strategii $\bar{\rho}$, která reprezentuje aktivní detektor chyb. Tato kapitola je rozdělena na dvě části. První část se zabývá přeformulováním úlohy, kdy je spojitá část stavu měřena přímo, neboli $\mathbf{y}_k = \mathbf{x}_k$. Druhá část je zaměřena na přeformulování úlohy pro systémy, kdy spojitá část stavu \mathbf{x}_k je měřena nepřímo skrz měřený výstup definovaný jako $\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mu_k) + \mathbf{H}_{\mu_k} \mathbf{v}_k$.

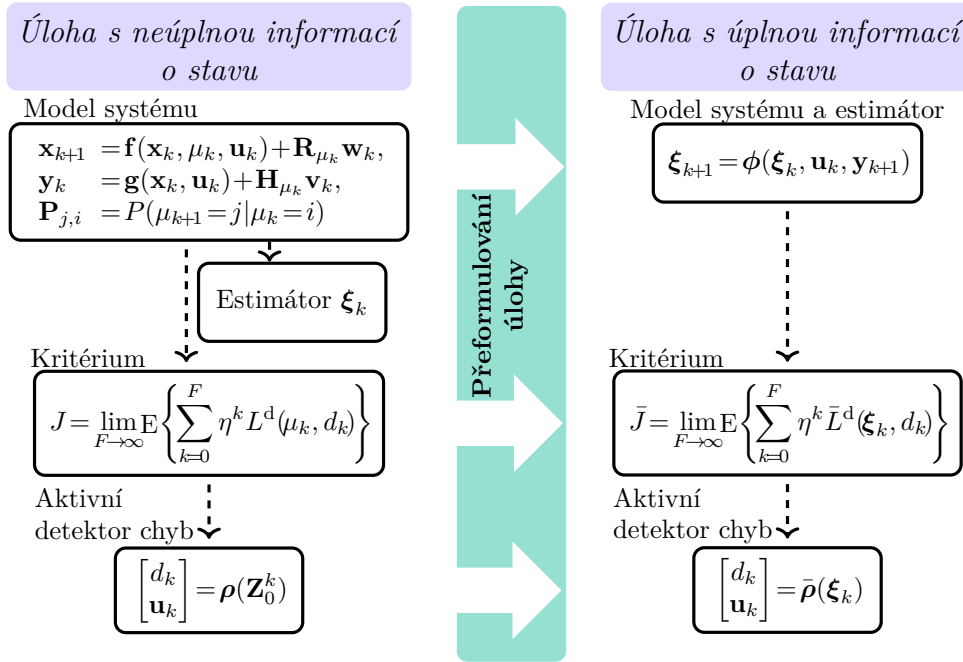
5.1.1 Úloha s přímo měřitelnou spojitou částí stavu

Přeformulování úlohy s přímo měřitelnou spojitou částí stavu je jednodušší než přeformulování úlohy s nepřímo měřitelnou spojitou částí stavu, z toho důvodu je uvedena dříve. Zároveň se využívá postačujících statistik, protože k jejich získání není nutné využít aproximační postup. Navržené přeformulování úlohy je v tomto případě exaktní i pro nelineární systémy.

Jelikož spojitá část stavu \mathbf{x}_k je známá, úkol spočívá ve vyjádření postačující statistiky poskytující informaci o budoucím indexu modelu μ_{k+1} . Necht $\mathbf{Y}_0^k = [(\mathbf{x}_0^k)^T, (\mathbf{u}_0^{k-1})^T]^T$ je informační vektor. Za postačující statistiku indexu modelu může být zvolena diskretní hustota pravděpodobnosti $P(\mu_k | \mathbf{Y}_0^k)$, která poskytuje informaci o podmíněných pravděpodobnostech indexů modelů. Necht je uvažován budoucí model $\mu_{k+1} = j, j \in \mathcal{M}$. Podmíněná pravděpodobnost budoucího indexu modelu za podmínky znalosti měření je definována jako

$$P(\mu_{k+1} = j | \mathbf{Y}_0^{k+1}) = \sum_{i=1}^{N_\mu} P(\mu_{k+1} = j | \mu_k = i | \mathbf{Y}_0^{k+1}) P(\mu_k = i | \mathbf{Y}_0^{k+1}), \quad (5.1)$$

kde $P(\mu_{k+1} = j | \mu_k = i | \mathbf{Y}_0^{k+1}) = P(\mu_{k+1} = j | \mu_k = i)$ je známá pravděpodobnost přechodu



Obrázek 5.2: Grafické znázornění přeformulování úlohy aktivní detekce chyb.

mezi příslušnými modely a $P(\mu_k = i | \mathbf{Y}_0^{k+1})$ je podmíněná pravděpodobnost, která může být spočítána jako

$$P(\mu_k = i | \mathbf{Y}_0^{k+1}) = \frac{p(\mathbf{x}_{k+1} | \mathbf{Y}_0^k, \mathbf{u}_k, \mu_k = i) P(\mu_k = i | \mathbf{Y}_0^k)}{p(\mathbf{x}_{k+1} | \mathbf{Y}_0^k, \mathbf{u}_k)}. \quad (5.2)$$

Prediktivní hustota pravděpodobnosti $p(\mathbf{x}_{k+1} | \mathbf{Y}_0^k, \mathbf{u}_k, \mu_k) = p_w(\mathbf{G}_{\mu_k}^{-1}(\mathbf{x}_{k+1} - \mathbf{f}(\mathbf{x}_k, \mu_k, \mathbf{u}_k)))$ je definována modelem (4.4). Podmíněná hustota pravděpodobnosti $p(\mathbf{x}_{k+1} | \mathbf{Y}_0^k, \mathbf{u}_k)$ může být dále vyjádřena jako suma čitatele (5.2) přes všechny možné modely, neboli

$$p(\mathbf{x}_{k+1} | \mathbf{Y}_0^k, \mathbf{u}_k) = \sum_{l=1}^{N_\mu} p(\mathbf{x}_{k+1} | \mathbf{Y}_0^k, \mathbf{u}_k, \mu_k = l) P(\mu_k = l | \mathbf{Y}_0^k). \quad (5.3)$$

Předpokládá se, že počáteční podmínky \mathbf{x}_0 a $P(\mu_0)$ jsou známé. Přímo měřitelná spojitá část stavu \mathbf{x}_k spolu s podmíněnou pravděpodobností indexu modelu $P(\mu_k | \mathbf{Y}_0^k)$ pak tvoří postačující statistiky stavu \mathbf{s}_k . V dalších podkapitolách bude dále předpokládáno, že spojitá část stavu \mathbf{x}_k není přímo měřitelná.

5.1.2 Lineární gaussovské systémy s nepřímo měřitelnou spojitou částí stavu

Tato část se zabývá získáním informačních statistik v úloze, kdy je systém popsán modelem (4.1), avšak funkce \mathbf{f} a \mathbf{g} mají speciální tvar

$$\begin{aligned}\mathbf{f}(\mathbf{x}_k, \mu_k, \mathbf{u}_k) &= \mathbf{A}_{\mu_k} \mathbf{x}_k + \mathbf{B}_{\mu_k} \mathbf{u}_k, \\ \mathbf{g}(\mathbf{x}_k, \mu_k) &= \mathbf{C}_{\mu_k} \mathbf{x}_k,\end{aligned}\tag{5.4}$$

kde \mathbf{A}_{μ_k} , \mathbf{B}_{μ_k} , \mathbf{C}_{μ_k} jsou známé matice pro všechny $\mu_k \in \mathcal{M}$. Zároveň se předpokládá, že \mathbf{w}_k a \mathbf{v}_k jsou vzájemně nezávislé bílé šумы s normovaným gaussovským rozdělením pravděpodobnosti, tj. $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n_x})$ a $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n_y})$.

Exaktní řešení úlohy odhadu stavu

V případě lineárních gaussovských systémů mají podmíněná prediktivní hustota pravděpodobnosti výstupu $p(\mathbf{y}_k | \mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1}, \mu_0^k)$ a podmíněná filtrační hustota pravděpodobnosti spojitě části stavu $p(\mathbf{x}_k | \mathbf{Z}_0^k, \mu_0^k)$ gaussovské rozdělení, které je jednoznačně definované střední hodnotou a kovarianční maticí. V důsledku je možné za informační statistiky stavu \mathbf{s}_k považovat podmíněné pravděpodobnosti posloupností modelů $P(\mu_0^k | \mathbf{Z}_0^k)$ spolu s prvními dvěma momenty spojitě části stavu. Hustota pravděpodobnosti $p(\mathbf{x}_k | \mathbf{Z}_0^k, \mu_0^k)$ může být vypočítána Kalmanovým filtrem (KF). To znamená, že pro časově variantní model odpovídající posloupnosti indexů modelů μ_0^k a použití banky KF mohou být vypočítány podmíněné pravděpodobnosti $P(\mu_0^k | \mathbf{Z}_0^k)$, střední hodnoty $\hat{\mathbf{x}}_{k|k}(\mu_0^k) = \mathbb{E}\{\mathbf{x}_k | \mathbf{Z}_0^k, \mu_0^k\}$ a kovarianční matice $\Sigma_{k|k}^x(\mu_0^k) = \mathbb{E}\{(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}(\mu_0^k))(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}(\mu_0^k))^T | \mathbf{Z}_0^k, \mu_0^k\}$.

Náhodná veličina \mathbf{y}_k definovaná podmíněnou prediktivní hustotou pravděpodobnosti $p(\mathbf{y}_k | \mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1}, \mu_0^k)$ má střední hodnotu $\hat{\mathbf{y}}_{k|k-1}(\mu_0^k)$ a kovarianční matici $\Sigma_{k|k-1}^y(\mu_0^k)$, které lze vypočítat jako

$$\hat{\mathbf{y}}_{k|k-1}(\mu_0^k) = \mathbf{C}_{\mu_k} \hat{\mathbf{x}}_{k|k-1}(\mu_0^{k-1}),\tag{5.5}$$

$$\Sigma_{k|k-1}^y(\mu_0^k) = \mathbf{C}_{\mu_k} \Sigma_{k|k-1}^x(\mu_0^{k-1}) \mathbf{C}_{\mu_k}^T + \mathbf{H}_{\mu_k} \mathbf{H}_{\mu_k}^T,\tag{5.6}$$

kde $\hat{\mathbf{x}}_{k|k-1}(\mu_0^{k-1})$ je podmíněná prediktivní střední hodnota \mathbf{x}_k , $\Sigma_{k|k-1}^x(\mu_0^{k-1})$ je podmíněná kovarianční matice \mathbf{x}_k a $\hat{\mathbf{x}}_{0|-1}$ spolu s $\Sigma_{0|-1}^x$ tvoří známou počáteční podmínku. Oba momenty charakterizující hustotu pravděpodobnosti $p(\mathbf{x}_k | \mathbf{Z}^{k-1}, \mathbf{u}_{k-1}, \mu_0^{k-1})$ lze vypočítat v kroku predikce stavu následovně

$$\hat{\mathbf{x}}_{k|k-1}(\mu_0^{k-1}) = \mathbf{A}_{\mu_{k-1}} \hat{\mathbf{x}}_{k-1|k-1}(\mu_0^{k-1}) + \mathbf{B}_{\mu_{k-1}} \mathbf{u}_{k-1},\tag{5.7}$$

$$\Sigma_{k|k-1}^x(\mu_0^{k-1}) = \mathbf{A}_{\mu_{k-1}} \Sigma_{k-1|k-1}^x(\mu_0^{k-1}) \mathbf{A}_{\mu_{k-1}}^T + \mathbf{G}_{\mu_{k-1}} \mathbf{G}_{\mu_{k-1}}^T.\tag{5.8}$$

Podmíněná filtrační hustota pravděpodobnosti spojitě části stavu $p(\mathbf{x}_k | \mathbf{Z}_0^k, \mu_0^k)$ je charakterizována střední hodnotou $\hat{\mathbf{x}}_{k|k}(\mu_0^k)$ a kovarianční maticí $\Sigma_{k|k}^x(\mu_0^k)$, které je možné získat

použitím následujících vztahů

$$\hat{\mathbf{x}}_{k|k}(\mu_0^k) = \hat{\mathbf{x}}_{k|k-1}(\mu_0^{k-1}) + \mathbf{K}(\mu_0^k) [\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}(\mu_0^k)], \quad (5.9)$$

$$\Sigma_{k|k}^x(\mu_0^k) = \Sigma_{k|k-1}^x(\mu_0^{k-1}) - \mathbf{K}(\mu_0^k) \mathbf{C}_{\mu_k} \Sigma_{k|k-1}^x(\mu_0^k), \quad (5.10)$$

kde $\mathbf{K}(\mu_0^k)$ je Kalmanův zisk definovaný jako

$$\mathbf{K}(\mu_0^k) = \Sigma_{k|k-1}^x(\mu_0^{k-1}) \mathbf{C}_{\mu_k}^T \Sigma_{k|k-1}^y(\mu_0^k)^{-1}. \quad (5.11)$$

Tímto postupem lze získat veškerou informaci o stavu \mathbf{s}_k . Informační statistiky $P(\mu_0^k | \mathbf{Z}_0^k)$, $\hat{\mathbf{x}}_{k|k}(\mu_0^k)$ a $\Sigma_{k|k}^x(\mu_0^k)$ jsou totiž zároveň postačujícími statistikami stavu \mathbf{s}_k , jelikož k jejich výpočtu není použita žádná aproximace. Nicméně výpočetní náročnost optimálního řešení úlohy odhadu stavu roste pro danou úlohu exponenciálně s časem v důsledku zvyšování počtu možných posloupností indexů modelů μ_0^k , jejichž počet odpovídá $(N_\mu)^k$. Proto je v následující části představeno možné aproximační řešení snižující výpočetní náročnost.

Suboptimální řešení úlohy odhadu stavu

Pro udržení zvládnutelné výpočetní náročnosti úlohy odhadu stavu mohou být využity různé suboptimální algoritmy jako zobecněný pseudo Bayesův (ZPB) algoritmus [86] nebo interagující vícemodelový algoritmus [87], které byly vyvinuty ve vědecké komunitě zabývající odhadem stavu. Následující část diskutuje použití ZPB algoritmu pro úlohu aktivní detekce chyb, který sleduje h -krokovou historii možných posloupností modelů. To znamená, že na začátku měření jsou sledovány všechny možné posloupnosti modelů a od časového okamžiku $k = h$ je sledováno pouze pevné množství posloupností modelů. V důsledku jsou pro všechny časové okamžiky $k \geq h$ posloupnosti modelů μ_{k-h}^k v rámci filtračního kroku algoritmu sloučeny tak, aby vytvořily sloučené posloupnosti μ_{k-h+1}^k . Výsledkem je omezení počtu všech možných posloupností indexů modelů na hodnotu $(N_\mu)^h$.

Podmíněné pravděpodobnosti posloupností modelů $P(\mu_0^k | \mathbf{Z}_0^k)$ mohou být použitím ZPB algoritmu aproximovány. Vztah (4.14) může být nahrazen vztahem

$$P(\mu_{k-h}^k | \mathbf{Z}_0^k) = \frac{p(\mathbf{y}_k | \mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1}, \mu_{k-h}^k) P(\mu_{k-h}^k | \mathbf{Z}_0^{k-1})}{p(\mathbf{y}_k | \mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1})}, \quad (5.12)$$

kde $P(\mu_{k-h}^k | \mathbf{Z}_0^{k-1})$ lze vyjádřit jako

$$P(\mu_{k-h}^k | \mathbf{Z}_0^{k-1}) = \begin{cases} P(\mu_0) & \text{if } k = 0, \\ P(\mu_k | \mu_{k-1}) P(\mu_{k-h}^{k-1} | \mathbf{Z}_0^{k-1}) & \text{if } k > 0. \end{cases} \quad (5.13)$$

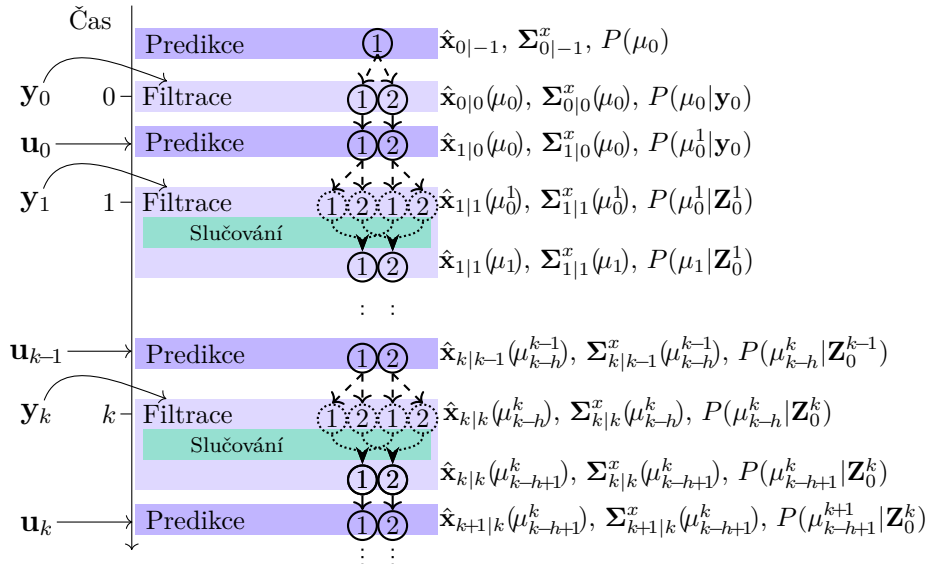
Postačující statistika \mathbf{s}_k ve formě podmíněné hustoty pravděpodobnosti $p(\mathbf{s}_k | \mathbf{Z}_0^k)$ je aproximována váženou sumou gaussovských hustot pravděpodobnosti s následujícími středními

hodnotami, kovariančními maticemi a vahami

$$\hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k) = \sum_{\mu_{k-h}^k} \frac{P(\mu_{k-h}^k | \mathbf{Z}_0^k)}{P(\mu_{k-h+1}^k | \mathbf{Z}_0^k)} \hat{\mathbf{x}}_{k|k}(\mu_{k-h}^k), \quad (5.14)$$

$$\Sigma_{k|k}^x(\mu_{k-h+1}^k) = \sum_{\mu_{k-h}^k} \frac{P(\mu_{k-h}^k | \mathbf{Z}_0^k)}{P(\mu_{k-h+1}^k | \mathbf{Z}_0^k)} \left(\hat{\mathbf{x}}_{k|k}(\mu_{k-h}^k) \hat{\mathbf{x}}_{k|k}^T(\mu_{k-h}^k) + \Sigma_{k|k}^x(\mu_{k-h}^k) \right) - \hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k) \hat{\mathbf{x}}_{k|k}^T(\mu_{k-h+1}^k), \quad (5.15)$$

$$P(\mu_{k-h+1}^k | \mathbf{Z}_0^k) = \sum_{\mu_{k-h}^k} P(\mu_{k-h}^k | \mathbf{Z}_0^k). \quad (5.16)$$



Obrázek 5.3: Ilustrace suboptimálního algoritmu odhadu stavu založeného na KF a ZPB pro systém s parametry $N_\mu = 2$, $h = 1$ a počátečním index modelu $\mu_0 = 1$.

Suboptimální algoritmus odhadu informačních statistik založený na KF a ZPB se skládá ze tří kroků: predikce, filtrace a slučování. Ilustrace algoritmu odhadu stavu je zobrazena na obrázku 5.3. Obrázek znázorňuje postup algoritmu odhadu informačních statistik v několika časových okamžicích. V kroku predikce jsou v závislosti na vstupu \mathbf{u}_k určeny budoucí hodnoty informačních statistik, které jsou následně aktualizovány na základě měření v kroku filtrace. Filtrace stavu zahrnuje slučování, které je provedeno porovnáním momentů, pomocí kterého jsou jednotlivé střední hodnoty $\hat{\mathbf{x}}_{k|k}(\mu_{k-h}^k)$, kovarianční matice $\Sigma_{k|k}^x(\mu_{k-h}^k)$ a podmíněné pravděpodobnosti posloupností modelů $P(\mu_{k-h}^k | \mathbf{Z}_0^k)$ sloučeny do $\hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k)$, $\Sigma_{k|k}^x(\mu_{k-h+1}^k)$ a $P(\mu_{k-h+1}^k | \mathbf{Z}_0^k)$.

Aproximační střední hodnota predikovaného výstupu $\hat{\mathbf{y}}_{k|k-1}(\mu_{k-h}^k)$ a příslušná kovarianční matice $\Sigma_{k|k-1}^y(\mu_{k-h}^k)$ mohou být vypočítány jako

$$\hat{\mathbf{y}}_{k|k-1}(\mu_{k-h}^k) = \mathbf{C}_{\mu_k} \hat{\mathbf{x}}_{k|k-1}(\mu_{k-h}^k), \quad (5.17)$$

$$\Sigma_{k|k-1}^y(\mu_{k-h}^k) = \mathbf{C}_{\mu_k} \Sigma_{k|k-1}^x(\mu_{k-h}^k) \mathbf{C}_{\mu_k}^T + \mathbf{H}_{\mu_k} \mathbf{H}_{\mu_k}^T. \quad (5.18)$$

Aproximační střední hodnota spojité části stavu $\hat{\mathbf{x}}_{k|k}(\mu_{k-h}^k)$ a kovarianční matice $\Sigma_{k|k}^x(\mu_{k-h}^k)$ jsou dány vztahy

$$\hat{\mathbf{x}}_{k|k}(\mu_{k-h}^k) = \hat{\mathbf{x}}_{k|k-1}(\mu_{k-h}^{k-1}) + \mathbf{K}(\mu_{k-h}^k)[\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}(\mu_{k-h}^k)], \quad (5.19)$$

$$\Sigma_{k|k}^x(\mu_{k-h}^k) = \Sigma_{k|k-1}^x(\mu_{k-h}^{k-1}) - \mathbf{K}(\mu_{k-h}^k)\mathbf{C}_{\mu_k}\Sigma_{k|k-1}^x(\mu_{k-h}^{k-1}), \quad (5.20)$$

$$\mathbf{K}(\mu_{k-h}^k) = \Sigma_{k|k-1}^x(\mu_{k-h}^{k-1})\mathbf{C}_{\mu_k}^T\Sigma_{k|k-1}^y(\mu_{k-h}^k)^{-1}. \quad (5.21)$$

Konečně, aproximační prediktivní střední hodnota spojité části stavu $\hat{\mathbf{x}}_{k+1|k}(\mu_{k-h+1}^k)$ a příslušná kovarianční matice $\Sigma_{k+1|k}^x(\mu_{k-h+1}^k)$ jsou dány následovně

$$\hat{\mathbf{x}}_{k+1|k}(\mu_{k-h+1}^k) = \mathbf{A}_{\mu_k}\hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k) + \mathbf{B}_{\mu_k}\mathbf{u}_k, \quad (5.22)$$

$$\Sigma_{k+1|k}^x(\mu_{k-h+1}^k) = \mathbf{A}_{\mu_k}\Sigma_{k|k}^x(\mu_{k-h+1}^k)\mathbf{A}_{\mu_k}^T + \mathbf{G}_{\mu_k}\mathbf{G}_{\mu_k}^T. \quad (5.23)$$

Informačními statistikami stavu \mathbf{s}_k jsou $\hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k)$, $\Sigma_{k|k}^x(\mu_{k-h+1}^k)$ a $P(\mu_{k-h+1}^k|\mathbf{Z}_0^k)$. Postup suboptimálního odhadu stavu založený na KF a ZPB je shrnutý v algoritmu 1. Informační statistiky jsou odhadnuty na základě kroků predikce a filtrace. Následně jsou informační statistiky použity ke generování rozhodnutí a budícího vstupu do systému na základě navrženého aktivního detektoru chyb. Tyto kroky se opakují v každém časovém okamžiku. Stojí za zmínku, že analogicky je možné místo ZPB algoritmu použít i některý další suboptimální algoritmus snižující výpočetní složitost spojenou s růstem počtu možných posloupností modelů μ_0^k v čase.

Algoritmus 1 Algoritmus odhadu informačních statistik založený na KF a ZPB pro úlohu aktivní detekce chyb.

Počáteční podmínky $P(\mu_0)$ a $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_{0|-1}, \Sigma_{0|-1}^x)$.

1. Predikce výstupu Výpočet $\hat{\mathbf{y}}_{k|k-1}$ a $\Sigma_{k|k-1}^y$ pomocí (5.17), resp. (5.18).

2. Pozorování Měření výstupu \mathbf{y}_k .

3. Filtrace Výpočet $\hat{\mathbf{x}}_{k|k}(\mu_{k-h}^k)$, $\Sigma_{k|k}^x(\mu_{k-h}^k)$ a $P(\mu_{k-h}^k|\mathbf{Z}_0^k)$ na základě (5.19), (5.20) a (5.12).

Porovnání momentů Výpočet $\hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k)$ a $\Sigma_{k|k}^x(\mu_{k-h+1}^k)$ pomocí (5.14) a (5.15). Výpočet $P(\mu_{k-h+1}^k|\mathbf{Z}_0^k)$ pomocí (5.16).

4. Rozhodnutí a vstup Generování rozhodnutí d_k a vstupu \mathbf{u}_k .

5. Predikce Výpočet $\hat{\mathbf{x}}_{k+1|k}$ a $\Sigma_{k+1|k}^x$ využitím vztahů (5.22) a (5.23).

Nastavení $k = k + 1$ a skok zpět na krok 1.

Pro kroky $k < h$ jsou statistiky $\hat{\mathbf{x}}_{k|k}$, $\Sigma_{k|k}^x$ a $P(\mu_{k-h+1}^k|\mathbf{Z}_0^k)$ vypočítány bez použití metody porovnání momentů, aby byly obdrženy statistiky pro všechny sledované posloupnosti indexů modelů.

5.1.3 Nelineární gaussovské systémy s nepřímo měřitelnou spojitou částí stavu

V této části se předpokládá, že chování systému může být popsáno modelem (4.1), kde \mathbf{f} a \mathbf{g} jsou nelineární vektorové funkce diferencovatelné vzhledem ke spojitě části stavu \mathbf{x}_k pro všechny možné indexy modelů a vstupy. Stavový šum $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n_x})$ a šum měření $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n_y})$ jsou vzájemně nezávislé bílé gaussovské šумы a \mathbf{I}_n je $n \times n$ jednotková matice. Na rozdíl od předchozího případu lineárních gaussovských systémů diskutovaného v podkapitole 5.1.2 není pro takto formulovanou úlohu řešení BRV známé. Stále se však zdá být vhodné zvolit jako informační statistiky první dva momenty spojitě části stavu \mathbf{x}_k a podmíněné pravděpodobnosti posloupnosti modelů $P(\mu_0^k | \mathbf{Z}_0^k)$. Tyto momenty mohou být spočítány některou metodou nelineárního odhadu stavu pro nelineární systémy s diferencovatelnými funkcemi \mathbf{f} a \mathbf{g} . Jednou z často používaných metod v úloze nelineárního odhadu stavu je rozšířený Kalmanův filtr (EKF), který bude využit v této části.

Suboptimální řešení úlohy odhadu stavu

V případě odhadu informačních statistik pomocí banky EKF a snižování stále rostoucího počtu sledovaných posloupností modelů pomocí ZPB algoritmu je postup analogický tomu, který je uveden v podkapitole 5.1.2. Celý algoritmus odhadu stavu se opět skládá z predikce, filtrace a slučování. Prediktivní střední hodnoty výstupu $\hat{\mathbf{y}}_{k|k-1}(\mu_{k-h}^k)$ a kovarianční matice $\Sigma_{k|k-1}^y(\mu_{k-h}^k)$ jsou spočítány jako

$$\hat{\mathbf{y}}_{k|k-1}(\mu_{k-h}^k) = \mathbf{g}(\hat{\mathbf{x}}_{k|k-1}(\mu_{k-h}^{k-1}), \mu_k), \quad (5.24)$$

$$\Sigma_{k|k-1}^y(\mu_{k-h}^k) = \nabla_g(\hat{\mathbf{x}}_{k|k-1}(\mu_{k-h}^{k-1}), \mu_k) \Sigma_{k|k-1}^x(\mu_{k-h}^{k-1}) \nabla_g^T(\hat{\mathbf{x}}_{k|k-1}(\mu_{k-h}^{k-1}), \mu_k) + \mathbf{H}_{\mu_k} \mathbf{H}_{\mu_k}^T, \quad (5.25)$$

kde $\nabla_g(\hat{\mathbf{x}}_{k|k-1}(\mu_{k-h}^{k-1}), \mu_k)$ je $n_y \times n_x$ Jacobiho matice, která je definována následovně

$$\nabla_g(\hat{\mathbf{x}}_{k|k-1}(\mu_{k-h}^{k-1}), \mu_k) = \left. \frac{\partial \mathbf{g}(\mathbf{x}_k, \mu_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1}(\mu_{k-h}^{k-1})}. \quad (5.26)$$

Filtrační střední hodnoty spojitě části stavu $\hat{\mathbf{x}}_{k|k}(\mu_{k-h}^k)$ a příslušné kovarianční matice $\Sigma_{k|k}^x(\mu_{k-h}^k)$ se vypočítají na základě následujících vztahů

$$\hat{\mathbf{x}}_{k|k}(\mu_{k-h}^k) = \hat{\mathbf{x}}_{k|k-1}(\mu_{k-h}^{k-1}) + \mathbf{K}(\mu_{k-h}^k) [\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}(\mu_{k-h}^k)], \quad (5.27)$$

$$\Sigma_{k|k}^x(\mu_{k-h}^k) = \Sigma_{k|k-1}^x(\mu_{k-h}^{k-1}) - \mathbf{K}(\mu_{k-h}^k) \nabla_g(\hat{\mathbf{x}}_{k|k-1}(\mu_{k-h}^{k-1}), \mu_k) \Sigma_{k|k-1}^x(\mu_{k-h}^{k-1}), \quad (5.28)$$

$$\mathbf{K}(\mu_{k-h}^k) = \hat{\Sigma}_{k|k-1}^x(\mu_{k-h}^{k-1}) \nabla_g^T(\hat{\mathbf{x}}_{k|k-1}(\mu_{k-h}^{k-1}), \mu_k) (\hat{\Sigma}_{k|k-1}^y(\mu_{k-h}^k))^{-1}. \quad (5.29)$$

Prediktivní střední hodnoty $\hat{\mathbf{x}}_{k+1|k}(\mu_{k-h+1}^k)$ a příslušné kovarianční matice $\Sigma_{k+1|k}^x(\mu_{k-h+1}^k)$ jsou vyjádřeny jako

$$\hat{\mathbf{x}}_{k+1|k}(\mu_{k-h+1}^k) = \mathbf{f}(\hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k), \mu_k, \mathbf{u}_k), \quad (5.30)$$

$$\Sigma_{k+1|k}^x(\mu_{k-h+1}^k) = \nabla_f(\hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k), \mu_k, \mathbf{u}_k) \Sigma_{k|k}^x(\mu_{k-h+1}^k) \nabla_f^T(\hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k), \mu_k, \mathbf{u}_k) + \mathbf{G}_{\mu_k} \mathbf{G}_{\mu_k}^T, \quad (5.31)$$

kde $\nabla_f(\hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k), \mu_k, \mathbf{u}_k)$ je $n_x \times n_x$ Jacobiho matice definována jako

$$\nabla_f(\hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k), \mu_k, \mathbf{u}_k) = \left. \frac{\partial \mathbf{f}(\mathbf{x}_k, \mu_k, \mathbf{u}_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k = \hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k)}. \quad (5.32)$$

Podmíněná hustota pravděpodobnosti $p(\mathbf{x}_k | \mathbf{Z}_0^k, \mu_0^k)$ je tedy aproximována gaussovským rozdělením pravděpodobnosti se střední hodnotou a kovariancí, které jsou vypočítány s využitím metody porovnání momentů ze vztahů (5.14), (5.15) a (5.16) na základě dílčích statistik generovaných EKF a ZPB. Řešení odhadu informačních statistik stavu \mathbf{s}_k založené na EKF a ZPB je shrnuté v algoritmu 2.

Algoritmus 2 Algoritmus odhadu informačních statistik založený na EKF a ZPB pro úlohu aktivní detekce chyb.

Počáteční podmínka $P(\mu_0)$ a $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_{0|-1}, \Sigma_{0|-1}^x)$.

1. Predikce výstupu Výpočet $\hat{\mathbf{y}}_{k|k-1}$ a $\Sigma_{k|k-1}^y$ pomocí (5.24), resp. (5.25).

2. Pozorování Měření výstupu \mathbf{y}_k .

3. Filtrace Výpočet $\hat{\mathbf{x}}_{k|k}(\mu_{k-h}^k)$, $\Sigma_{k|k}^x(\mu_{k-h}^k)$ a $P(\mu_{k-h}^k | \mathbf{Z}_0^k)$ na základě (5.27), (5.28) a (5.12).

Porovnání momentů Výpočet $\hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k)$ a $\Sigma_{k|k}^x(\mu_{k-h+1}^k)$ pomocí (5.14) a (5.15). Výpočet $P(\mu_{k-h+1}^k | \mathbf{Z}_0^k)$ pomocí (5.16).

4. Rozhodnutí a vstup Generování rozhodnutí d_k a vstupu \mathbf{u}_k .

5. Predikce Výpočet $\hat{\mathbf{x}}_{k+1|k}$ a $\Sigma_{k+1|k}^x$ využitím vztahů (5.30) a (5.31).

Nastavení $k = k + 1$ a skok zpět na krok 1.

Pro kroky $k < h$ jsou statistiky $\hat{\mathbf{x}}_{k|k}$, $\Sigma_{k|k}^x$ a $P(\mu_{k-h+1}^k | \mathbf{Z}_0^k)$ vypočítány bez metody porovnání momentů, aby byly obdrženy statistiky pro všechny sledované posloupnosti indexů modelů.

5.1.4 Nelineární negaussovské systémy s nepřímo měřitelnou spojitou částí stavu

Tato část se věnuje obecnému případu, kdy chování systému může být popsáno modelem (4.1), kde \mathbf{f} a \mathbf{g} jsou nelineární vektorové funkce. To zahrnuje i případy, kdy \mathbf{f} a \mathbf{g} nejsou diferencovatelné. Stavový šum $\mathbf{w}_k \in \mathbb{R}^{n_x}$ a šum měření $\mathbf{v}_k \in \mathbb{R}^{n_y}$ jsou vzájemně nezávislé bílé šумы popsané známými hustotami pravděpodobnostmi p_{w_k} , resp. p_{v_k} . Počáteční stav \mathbf{x}_0 může být taktéž generovaný z libovolného předem známého rozdělení pravděpodobnosti p_{x_0} . Jelikož se EKF nehodí použít pro systémy se silně nelineárními nebo nediferencovatelnými funkcemi \mathbf{f} a \mathbf{g} , navržené řešení využívá částicový filtr (PF).

Suboptimální řešení úlohy odhadu stavu

K odhadu informačních statistik pro danou třídu systémů je zvolen bootstrap PF [88]. Oproti technikám založeným na KF je PF méně omezující z hlediska požadavků na model. Hlavní myšlenkou PF je aproximovat podmíněnou hustotu pravděpodobnosti stavu

pomocí empirické hustoty pravděpodobnosti, která využívá náhodné vzorky, zvané částice, a přidružené váhy. Částicový filtr postupně generuje nové částice a aktualizuje přidružené váhy na základě měřeného výstupu systému. Významnými faktory, které ovlivňují kvalitu filtrace jsou počet částic N_p a způsob převzorkování. Převzorkování pomáhá eliminovat situaci, kdy se téměř všechny váhy stanou nulové a pouze jedna váha rovna jedné [89]. Na druhou stranu při převzorkování dochází k degeneraci částic a ztrátě informace. Z důvodu zjednodušení návrhu estimátoru stavu v úloze aktivní detekce chyb bude použit PF s pevným počtem částic a multinomiálním převzorkováním, které je provedeno pokaždé aktualizací vah [88]. Rozšířené varianty PF s proměnným počtem částic a efektivnějším způsobem převzorkování mohou být nalezeny například v [89]–[91]. Pro systémy se skokovými změnami chování byl PF uveden v [92].

Podmíněnou hustotu pravděpodobnosti $p(\mathbf{s}_k|\mathbf{Z}_0^k)$ je možné aproximovat pomocí empirické hustoty pravděpodobnosti $\pi(\mathbf{s}_k|\mathbf{Z}_0^k)$ jako

$$p(\mathbf{s}_k|\mathbf{Z}_0^k) \approx \pi(\mathbf{s}_k|\mathbf{Z}_0^k) = \sum_{i=1}^{N_p} v_k^{(i)} \delta(\mathbf{s}_k - \mathbf{s}_k^{(i)}), \quad (5.33)$$

kde $\mathbf{s}_k^{(i)} \in \mathcal{S}$ je částice, $v_k^{(i)} \in \mathbb{R}^+$ je váha příslušící k dané částici, N_p je počet částic a δ je Diracova delta funkce. Váhy splňují podmínku $\sum_{i=1}^{N_p} v_k^{(i)} = 1$. Empirická hustota pravděpodobnosti $\pi(\mathbf{s}_k|\mathbf{Z}_0^k)$ plně popisuje $p(\mathbf{s}_k|\mathbf{Z}_0^k)$, jestliže $N_p \rightarrow \infty$. Samotný odhad stavu se skládá ze dvou kroků: predikce a filtrace. V kroku predikce jsou generovány nové částice $\{\mathbf{s}_{k+1}^{(i)}\}_{i=1}^{N_p}$ na základě minulých částic $\{\mathbf{s}_k^{(i)}\}_{i=1}^{N_p}$ a modelu systému (4.1) a (4.3). Nejprve je z hustoty pravděpodobnosti p_{w_k} vzorkováno N_p i.i.d. částic $\{\mathbf{w}_k^{(i)}\}_{i=1}^{N_p}$, které jsou následně propagovány skrz model (4.1) tak, aby byly získány částice $\{\mathbf{x}_{k+1}^{(i)}\}_{i=1}^{N_p}$. Následně se vygenerují částice $\{\mu_{k+1}^{(i)}\}_{i=1}^{N_p}$ podle známých pravděpodobností přechodů (4.3). Částice stavu jsou poté vytvořeny jako $\mathbf{s}_{k+1}^{(i)} = [(\mathbf{x}_{k+1}^{(i)})^T, \mu_{k+1}^{(i)}]^T$, $i = 1, 2, \dots, N_p$. V kroku filtrace jsou váhy příslušných částic stavu aktualizovány na základě měřeného výstupu \mathbf{y}_k podle vztahu

$$v_k^{(i)} = \frac{p(\mathbf{y}_k|\mathbf{s}_k^{(i)}, \mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1})}{\sum_{j=1}^{N_p} p(\mathbf{y}_k|\mathbf{s}_k^{(j)}, \mathbf{Z}_0^{k-1}, \mathbf{u}_{k-1})}, \quad i = 1, \dots, N_p. \quad (5.34)$$

Aby bylo zabráněno degeneraci vah s časem, částice stavu s nízkou vahou jsou v rámci převzorkování eliminovány a částice s vysokou vahou jsou duplikovány. Po aplikaci multinomiálního převzorkování jsou získány nové částice stavu $\{\mathbf{s}_k^{*(i)}\}_{i=1}^{N_p}$ s normalizovanými vahami $v_k^{*(i)} = 1/N_p$, $i = 1, \dots, N_p$. V důsledku je empirická hustota pravděpodobnosti $\pi(\mathbf{s}_k|\mathbf{Z}_0^k)$ zjednodušena do tvaru

$$\pi(\mathbf{s}_k|\mathbf{Z}_0^k) = \frac{1}{N_p} \sum_{i=1}^{N_p} \delta(\mathbf{s}_k - \mathbf{s}_k^{(i)}). \quad (5.35)$$

Informační statistikou pro stav \mathbf{s}_k je empirická hustota pravděpodobnosti $\pi(\mathbf{s}_k|\mathbf{Z}_0^k)$, kterou tvoří částice a jejich váhy.

Je patrné, že dimenze takové informační statistiky může být velmi vysoká díky přímé závislosti na počtu vzorků N_p . Právě vysoká dimenze by mohla způsobit ohromné výpočetní nároky při řešení optimalizační úlohy. Proto je vhodné nejprve provést redukcii dimenze, například odhadem prvních dvou momentů spojitě části stavu \mathbf{x}_k a podmíněné pravděpodobnosti indexů modelů podobně, jak je uvedeno v podkapitolách 5.1.2 a 5.1.3.

Nechť je definována množina indexů částic jako

$$\mathcal{P}_j = \{i \in \{1, 2, \dots, N_p\} : \mu_k^{(i)} = j\} \quad (5.36)$$

pro všechna $j \in \mathcal{M}$. Množinu \mathcal{P}_j tvoří všechny indexy z množiny $\{1, 2, \dots, N_p\}$, pro které poslední složka částice stavu, neboli $\mu_k^{(i)}$, nabývá hodnoty j . Podmíněná pravděpodobnost pro jednotlivé indexy modelu j může být aproximačně vypočítána jako

$$P(\mu_k = j | \mathbf{Z}_0^k) = \sum_{i \in \mathcal{P}_j} v_k^{(i)}, \quad (5.37)$$

Aproximativní střední hodnota spojitě části stavu \mathbf{x}_k podmíněná informačním vektorem \mathbf{Z}_0^k a indexem modelu j je vypočítána jako

$$\hat{\mathbf{x}}_{k|k}(j) = \frac{\sum_{i \in \mathcal{P}_j} v_k^{(i)} \mathbf{x}_k^{(i)}}{P(\mu_k = j | \mathbf{Z}_0^k)} \quad (5.38)$$

a odpovídající aproximační kovarianční matice jako

$$\Sigma_{k|k}^x(j) = \frac{\sum_{i \in \mathcal{P}_j} v_k^{(i)} \mathbf{x}_k^{(i)} (\mathbf{x}_k^{(i)})^T}{P(\mu_k = j | \mathbf{Z}_0^k)} - \hat{\mathbf{x}}_{k|k}(j) (\hat{\mathbf{x}}_{k|k}(j))^T. \quad (5.39)$$

Jestliže $|\mathcal{P}_j| = 0$, pak hodnoty prvních dvou momentů mohou být zvoleny libovolně, jelikož nebudou v dalších výpočtech uvažovány. Informačními statistikami stavu \mathbf{s}_k jsou $P(\mu_k | \mathbf{Z}_0^k)$, $\hat{\mathbf{x}}_{k|k}(\mu_k)$ a $\Sigma_{k|k}^x(\mu_k)$. Postup odhadu informačních statistik na základě bootstrap PF je shrnutý v algoritmu 3. Stojí za zmínku, že převzorkování je z důvodu přesnosti záměrně provedeno až po vytvoření informačních statistik, jelikož při převzorkování dochází ke ztrátě informace. Zároveň je zapotřebí si uvědomit, že PF je stochastický filtr, jelikož pro odhad stavu používá náhodné vzorky šumu a náhodné vzorky převzorkování. To znamená, že na rozdíl od KF pro stejné vstupy může PF generovat různé výstupy.

5.2 Postup přeformulování úlohy

V předchozí podkapitole 5.1 bylo přiblíženo téma informačních statistik, které jsou potřebné k samotnému přeformulování úlohy s neúplnou informací o stavu na úlohu s úplnou informací o stavu. Přeformulování úlohy spočívá v transformaci původního modelu systému, návrhového kritéria a aktivního detektoru chyb tak, aby závisely na informačních statistikách. Blokový diagram ilustrující úlohu aktivní detekce chyb po přeformulování je zobrazen na obrázku 5.4. Blokový diagram znázorňuje vazby mezi systémem a aktivním

Algoritmus 3 Algoritmus odhadu informačních statistik založený na bootstrap PF pro úlohu aktivní detekce chyb.

Počáteční podmínky $P(\mu_0)$, $p(\mathbf{x}_0)$ a počet částic N_p .

1. Inicializace Generování N_p i.i.d. částic stavu $\{\mathbf{s}_0^{(i)}\}_{i=1}^{N_p}$ z počátečního rozdělení pravděpodobnosti $p(\mathbf{s}_0) = p(\mathbf{x}_0)P(\mu_0)$.

2. Pozorování

Měření výstupu \mathbf{y}_k .

3. Filtrace

3.1 Výpočet vah $\{v_k^{(i)}\}_{i=1}^{N_p}$ na základě (5.34).

3.2 Empirická hustota pravděpodobnosti $\pi(\mathbf{s}_k|\mathbf{Z}_0^k)$ je pak dána (5.33).

4. Redukce dimenze

Výpočet podmíněných pravděpodobností indexů modelů $P(\mu_k|\mathbf{Z}_0^k)$ a prvních dvou momentů spojité části stavu $\hat{\mathbf{x}}_{k|k}(\mu_k)$ a $\Sigma_{k|k}^x(\mu_k)$ na základě (5.37), (5.38), resp. (5.39).

5. Rozhodnutí a vstup

Generování rozhodnutí d_k a vstupu \mathbf{u}_k .

6. Převzorkování

Nové částice $\{\mathbf{s}_k^{*(j)}\}_{j=1}^{N_p}$ jsou získány multinomiálním převzorkováním s diskretním rozdělením $P(\mathbf{s}_k^* = \mathbf{s}_k^{(i)}) = v_k^{(i)}$. Po převzorkování příslušných částic $\{\mathbf{s}_k^{*(j)}\}_{j=1}^{N_p}$ mají všechny váhy hodnotu $v_k^{*(j)} = \frac{1}{N_p}$ pro $j = 1, \dots, N_p$.

7. Predikce

7.1 Generování náhodných částic $\mathbf{w}_k^{(i)}$ z hustoty pravděpodobnosti stavového šumu p_w pro $i = 1, \dots, N_p$.

7.2 Generování náhodných částic $\mu_{k+1}^{(i)}$ z diskretního rozdělení $P(\mu_{k+1}|\mu_k = \mu_k^{*(i)})$ pro $i = 1, \dots, N_p$.

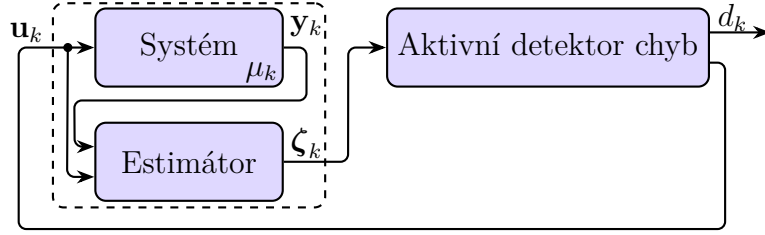
7.3 Výpočet částic $\mathbf{x}_{k+1}^{(i)}$ pro $i = 1, \dots, N_p$ následovně

$$\mathbf{x}_{k+1}^{(i)} = \mathbf{f}(\mathbf{x}_k^{*(i)}, \mu_k^{*(i)}, \mathbf{u}_k) + \mathbf{R}_{\mu_k^{*(i)}} \mathbf{w}_k^{(i)}. \quad (5.40)$$

7.4 Sestavení částic stavu $\mathbf{s}_{k+1}^{(i)} = [(\mathbf{x}_{k+1}^{(i)})^T, \mu_{k+1}^{(i)}]^T$ pro $i = 1, \dots, N_p$.

Nastavení $k = k + 1$ a skok zpět na krok 2.

detektorem chyb. Estimátor na základě dostupné informace odhadne informační statistiky stavu \mathbf{s}_k , které jsou předány do aktivního detektoru chyb ve formě vektoru informačních statistik $\boldsymbol{\xi}_k$. Tento vektor je v práci nazván jako *hyperstav*. Na základě hyperstavu generuje aktivní detektor chyb rozhodnutí o chybě d_k spolu se vstupním signálem \mathbf{u}_k . Stojí za zmínku, že v obecném případě systémů s nepřímo měřitelnou spojitou částí stavu je hyperstav označen symbolem $\boldsymbol{\xi}_k$ zatímco v případě systémů s přímo měřitelnou spojitou částí stavu je hyperstav označen jako $\boldsymbol{\zeta}_k$ z důvodu rozdílné struktury informačních statistik. V případě $\boldsymbol{\zeta}_k$ se zároveň jedná o postačující statistiky stavu \mathbf{s}_k .



Obrázek 5.4: Blokový diagram úlohy aktivní detekce chyb po přeformulování na úlohu s úplnou informací o stavu.

5.2.1 Úloha s přímo měřitelnou spojitou částí stavu

Nechť je definován hyperstav ζ_k jako vektor obsahující postačující statistiky stavu \mathbf{s}_k ve formě přímo měřitelné spojitě části stavu \mathbf{x}_k a podmíněných pravděpodobností modelů $P(\mu_k = i | \mathbf{Y}_0^k)$, $i = 1, 2, \dots, N_\mu$, neboli

$$\zeta_k = [\mathbf{x}_k^T, P(\mu_k = 1 | \mathbf{Y}_0^k), P(\mu_k = 2 | \mathbf{Y}_0^k) \dots, P(\mu_k = N_\mu - 1 | \mathbf{Y}_0^k)]^T, \quad (5.41)$$

kde $\zeta_k \in \mathcal{R} = \mathbb{R}^{n_x} \times \mathcal{B} \subset \mathbb{R}^{n_\zeta}$ je hyperstav z množiny přípustných hodnot \mathcal{R} , která je definována pomocí množiny $\mathcal{B} = \{\mathbf{b} \in \mathbb{R}^{N_\mu-1} : \mathbf{b} \geq \mathbf{0}, \mathbf{1}^T \mathbf{b} \leq 1\}$, kde $\mathbf{0}$ a $\mathbf{1}$ jsou vektory nul, resp. jedniček, příslušných dimenzí. Vektor hyperstavu záměrně obsahuje pouze $N_\mu - 1$ podmíněných pravděpodobností indexů modelů z důvodu snížení dimenze, jelikož platí $\sum_{i=1}^{N_\mu} P(\mu_k = i | \mathbf{Y}_0^k) = 1$, neboli zbývající podmíněná pravděpodobnost může být snadno vypočítána jako

$$P(\mu_k = N_\mu | \mathbf{Y}_0^k) = 1 - \sum_{i=1}^{N_\mu-1} P(\mu_k = i | \mathbf{Y}_0^k). \quad (5.42)$$

Dimenze hyperstavu je rovna hodnotě $n_\zeta = n_x + N_\mu - 1$.

Původní úloha definovaná v podkapitole 4.1 pro systémy s přímo měřitelnou spojitou částí stavu může být přeformulována pomocí hyperstavu ζ_k na úlohu s úplnou informací o stavu. Takový postup zahrnuje transformaci původního modelu systému (4.4), aktivního detektoru chyb (4.6) a kritéria (4.7) na nový model systému, aktivní detektor chyb a kritérium. Nový model popisující chování systému v každém časovém okamžiku $k \in \mathcal{T}$ může být definován jako

$$\zeta_{k+1} = \phi(\zeta_k, \mathbf{u}_k, \mathbf{x}_{k+1}), \quad (5.43)$$

kde $\phi : \mathcal{R} \times \mathcal{U} \times \mathbb{R}^{n_x} \mapsto \mathcal{R}$ je nelineární vektorová funkce popisující vývoj hyperstavu ζ_k v čase. Na funkci ϕ lze však pohlížet také jako na funkci popisující odhad postačujících statistik stavu \mathbf{s}_k a chování systému dané původním modelem. Spojitá část stavu \mathbf{x}_{k+1} je tedy dána modelem (4.4).

Aktivní detektor chyb je definován pomocí stacionární strategie, která závisí na hyperstavu ζ_k . Rozhodnutí o chybě v systému a budicí vstupní signál jsou tak generovány na

základě

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \bar{\rho}(\zeta_k) = \begin{bmatrix} \bar{\sigma}(\zeta_k) \\ \bar{\gamma}(\zeta_k) \end{bmatrix}, \quad (5.44)$$

kde $\bar{\rho} : \mathcal{R} \mapsto \mathcal{M} \times \mathcal{U}$, $\bar{\sigma} : \mathcal{R} \mapsto \mathcal{M}$ a $\bar{\gamma} : \mathcal{R} \mapsto \mathcal{U}$ jsou příslušné strategie aktivního detektoru chyb, generátoru rozhodnutí a generátoru budicího vstupního signálu.

Původní kritérium (4.7) je zapotřebí transformovat tak, aby bylo závislé na hyperstavu ζ_k , avšak zůstalo ekvivalentní k původním požadavkům. Kritérium může být přepsáno do následujícího tvaru

$$\bar{J} = \lim_{F \rightarrow \infty} \mathbb{E} \left\{ \sum_{k=0}^F \eta^k \bar{L}^d(\zeta_k, d_k) \right\}, \quad (5.45)$$

kde $\bar{L}^d : \mathcal{R} \times \mathcal{M} \mapsto \mathbb{R}^+$ je funkce ekvivalentní k L^d definovaná jako

$$\bar{L}^d(\zeta_k, d_k) = \mathbb{E} \left\{ L^d(\mu_k, d_k) | \mathbf{Y}_0^k, d_k \right\}. \quad (5.46)$$

Jelikož index modelu μ_k je diskrétní náhodná veličina a hyperstav ζ_k obsahuje postačující statistiky stavu \mathbf{s}_k , může být funkce (5.46) upravena do následujícího tvaru

$$\bar{L}^d(\zeta_k, d_k) = (\mathbf{1}(d_k))^T \boldsymbol{\varrho}(\zeta_k), \quad (5.47)$$

kde $\mathbf{1}(d_k) = [L^d(1, d_k), \dots, L^d(N_\mu, d_k)]^T$, $\mathbf{1} : \mathcal{M} \mapsto (\mathbb{R}^+)^{N_\mu}$, je funkce definující vektor jednotlivých ztrát odpovídající rozhodnutí d_k a $\boldsymbol{\varrho}(\zeta_k) = [P(\mu_k = 1 | \mathbf{Y}_0^k), \dots, P(\mu_k = N_\mu | \mathbf{Y}_0^k)]^T$, $\boldsymbol{\varrho} : \mathcal{R} \mapsto \mathbb{R}^{N_\mu}$, je funkce definující vektor jednotlivých podmíněných pravděpodobností indexů modelů. V důsledku mohou být ztráty vypočteny vynásobením těchto dvou vektorů, jak bylo uvedeno v (5.47).

Úloha aktivní detekce chyb a řízení

V případě úlohy návrhu aktivního detektoru chyb a řízení nový model popisující chování systému odpovídá (5.43). Aktivní detektor chyb a regulátor lze definovat pomocí stacionární strategie závislé na hyperstavu ζ_k jako

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \bar{\rho}^r(\zeta_k) = \begin{bmatrix} \bar{\sigma}^r(\zeta_k) \\ \bar{\gamma}^r(\zeta_k) \end{bmatrix}, \quad (5.48)$$

kde $\bar{\rho}^r : \mathcal{R} \mapsto \mathcal{M} \times \mathcal{U}$, $\bar{\sigma}^r : \mathcal{R} \mapsto \mathcal{M}$ a $\bar{\gamma}^r : \mathcal{R} \mapsto \mathcal{U}$ jsou hledané strategie. Návrhové požadavky definované v původním kritériu (4.19) mohou být definovány v následujícím kritériem jako

$$\bar{J} = \lim_{F \rightarrow \infty} \mathbb{E} \left\{ \sum_{k=0}^F \eta^k \bar{L}(\zeta_k, d_k, \mathbf{u}_k) \right\}, \quad (5.49)$$

kde $\bar{L} : \mathcal{R} \times \mathcal{M} \times \mathcal{U} \mapsto \mathbb{R}^+$ je ztrátová funkce ekvivalentní původní funkci L , která se opět skládá ze dvou částí hodnotících kvalitu detekce a řízení

$$\bar{L}(\zeta_k, d_k, \mathbf{u}_k) = \alpha \bar{L}^d(\zeta_k, d_k) + (1 - \alpha) \bar{L}^c(\zeta_k, \mathbf{u}_k), \quad (5.50)$$

přičemž jednotlivé funkce $\bar{L}^d : \mathcal{R} \times \mathcal{M} \mapsto \mathbb{R}^+$ a $\bar{L}^c : \mathcal{R} \times \mathcal{U} \mapsto \mathbb{R}^+$ jsou definovány následovně

$$\bar{L}^d(\zeta_k, d_k) = \mathbb{E} \left\{ L^d(\mu_k, d_k) | \mathbf{Y}_0^k, d_k \right\}, \quad (5.51)$$

$$\bar{L}^c(\zeta_k, \mathbf{u}_k) = L^c \left([\zeta_{k,1}, \zeta_{k,2}, \dots, \zeta_{k,n_\xi}]^T, \mathbf{u}_k \right). \quad (5.52)$$

Pohledem na (5.52) je patrné, že přeformulovaná ztrátová funkce řízení \bar{L}^c odpovídá té původní z důvodu přímé měřitelnosti spojitě části stavu \mathbf{x}_k . Vektor $[\zeta_{k,1}, \zeta_{k,2}, \dots, \zeta_{k,n_\xi}]^T = \mathbf{x}_k$ tedy odpovídá přímo měřitelné spojitě části stavu.

Tímto byla popsána transformace původní optimalizační úlohy na úlohu s úplnou informací o stavu pro systémy s přímo měřitelnou spojitou částí stavu \mathbf{x}_k . Následující část se týká obecnější úlohy, kdy spojitá část stavu není přímo měřitelná. Rozdílem oproti této části je použití informačních statistik ve formě podmíněných pravděpodobností indexů modelů a prvních dvou momentů spojitě části stavu.

5.2.2 Úloha s nepřímo měřitelnou spojitou částí stavu

Tato část se věnuje přeformulování původní úlohy definované v podkapitole 4.1 pro systémy s nepřímo měřitelným \mathbf{x}_k na úlohu s úplnou informací o stavu. Z důvodu obecnosti jsou zvoleny informační statistiky ve formě středních hodnot $\hat{\mathbf{x}}_{k|k}(\mu_{k-h+1}^k)$, kovariančních matic $\Sigma_{k|k}^x(\mu_{k-h+1}^k)$ a podmíněných pravděpodobností posloupností modelů $P(\mu_{k-h+1}^k | \mathbf{Z}_0^k)$, které jsou uvedeny v podkapitolách 5.1.2 a 5.1.3. Analogický postup definice hyperstavu a přeformulování úlohy může být proveden také pro statistiky $\hat{\mathbf{x}}_{k|k}(\mu_k)$, $\Sigma_{k|k}^x(\mu_k)$ a $P(\mu_k | \mathbf{Z}_0^k)$ z podkapitoly 5.1.4.

Nechť $\xi_k \in \mathcal{G} \subset \mathbb{R}^{n_\xi}$ je hyperstav definovaný jako

$$\xi_k = [\hat{\mathbf{x}}_{k|k}^T((\mu_{k-h+1}^k)_1), \dots, \hat{\mathbf{x}}_{k|k}^T((\mu_{k-h+1}^k)_{N_{\text{ms}}}), \Gamma\{\Sigma_{k|k}^x((\mu_{k-h+1}^k)_1)\}^T, \dots, \Gamma\{\Sigma_{k|k}^x((\mu_{k-h+1}^k)_{N_{\text{ms}}})\}^T, P((\mu_{k-h+1}^k)_1 | \mathbf{Z}_0^k), \dots, P((\mu_{k-h+1}^k)_{N_{\text{ms}}-1} | \mathbf{Z}_0^k)]^T, \quad (5.53)$$

kde značení $(\mu_{k-h+1}^k)_i$ odpovídá i -té posloupnosti modelů μ_{k-h+1}^k , $i = \{1, 2, \dots, N_{\text{ms}}\}$, N_{ms} je celkový počet posloupností modelů, a dále $\Gamma\{\cdot\}$ je operátor, který vrací dolní trojúhelníkovou matici svého argumentu a ve vztahu (5.53) je použitý z důvodu redukce dimenze hyperstavu. Jeho použití dovoluje symetrie kovariančních matic $\Sigma_{k|k}^x((\mu_{k-h+1}^k)_i)$. Redukce stavu je taktéž dosažena vynecháním jedné pravděpodobnosti posloupnosti modelů, podobně jako v (5.41). Dimenze hyperstavu je rovna $n_\xi = N_{\text{ms}}(n_x^2 + 3n_x + 2)/2 - 1$. Kovarianční matice jsou do hyperstavu zahrnuty z důvodu zachování částečné informace o kvalitě odhadu.

S využitím hyperstavu ξ_k může být původní systém (4.1) přepsán do následujícího nelineárního modelu popisujícího vývoj hyperstavu

$$\xi_{k+1} = \phi(\xi_k, \mathbf{u}_k, \mathbf{y}_{k+1}), \quad (5.54)$$

kde $\phi : \mathcal{G} \times \mathcal{U} \times \mathbb{R}^{n_y} \mapsto \mathcal{G}$ je nelineární vektorová funkce popisující chování původního systému spojeného s algoritmem odhadu informačních statistik stavu \mathbf{s}_k a na budoucí výstup \mathbf{y}_{k+1} je pohlíženo jako na náhodnou veličinu s prediktivní hustotou pravděpodobnosti $p(\mathbf{y}_{k+1} | \mathbf{Z}_0^k, \mathbf{u}_k)$. Momenty náhodné veličiny \mathbf{y}_{k+1} mohou být odhadnuty příslušnou estimační metodou, například pomocí KF nebo EKF s ZPB algoritmem nebo pomocí PF.

Aktivní detektor chyb (4.5) může být přeformulován jako

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \bar{\rho}(\boldsymbol{\xi}_k) = \begin{bmatrix} \bar{\sigma}(\boldsymbol{\xi}_k) \\ \bar{\gamma}(\boldsymbol{\xi}_k) \end{bmatrix}, \quad (5.55)$$

kde $\bar{\rho} : \mathcal{G} \mapsto \mathcal{M} \times \mathcal{U}$, $\bar{\sigma} : \mathcal{G} \mapsto \mathcal{M}$ a $\bar{\gamma} : \mathcal{G} \mapsto \mathcal{U}$ jsou stacionární strategie popisující v daném pořadí aktivní detektor chyb, generátor rozhodnutí a generátor vstupního signálu. Cílem je najít takovou strategii $\bar{\rho}$, která minimalizuje ztráty způsobené chybnými rozhodnutími o chování v systému.

Původní kritérium kvality detekce chyb (4.7) může být přeformulováno jako

$$\bar{J} = \lim_{F \rightarrow \infty} \mathbb{E} \left\{ \sum_{k=0}^F \eta^k \bar{L}^d(\boldsymbol{\xi}_k, d_k) \right\}, \quad (5.56)$$

kde $\bar{L}^d : \mathcal{G} \times \mathcal{M} \mapsto \mathbb{R}^+$ je funkce penalizující chybná rozhodnutí, která odpovídá původní funkci L^d a je definována jako

$$\bar{L}^d(\boldsymbol{\xi}_k, d_k) \approx \mathbb{E} \left\{ L^d(\mu_k, d_k) | \mathbf{Z}_0^k, d_k \right\}. \quad (5.57)$$

V případě celé sledované historie $h = k + 1$ je možné ztrátovou funkci \bar{L}^d definovanou v (5.57) přepsat do následujícího tvaru

$$\bar{L}^d(\boldsymbol{\xi}_k, d_k) = (\mathbf{1}(d_k))^T \mathbf{M} \boldsymbol{\rho}(\boldsymbol{\xi}_k), \quad (5.58)$$

kde $\mathbf{M} \in \mathbb{R}^{N_\mu \times N_{ms}}$ je matice nul a jedniček, která sčítá pravděpodobnosti posloupností modelů končících ve stejném modelu a funkce $\boldsymbol{\rho}$ je oproti vztahu (5.47) definovaná jako $\boldsymbol{\rho}(\boldsymbol{\xi}_k) = [\hat{P}((\mu_{k-h+1}^k)_1 | \mathbf{Z}_0^k), \dots, \hat{P}((\mu_{k-h+1}^k)_{N_{ms}} | \mathbf{Z}_0^k)]^T$, $\boldsymbol{\rho} : \mathcal{G} \mapsto \mathbb{R}^{N_\mu}$. Jestliže platí $h < k$, neboli dochází k určité ztrátě informace o stavu \mathbf{s}_k , vztah (5.58) je pouhou aproximací ztrátové funkce detekce ze vztahu (5.57).

Úloha aktivní detekce chyb a řízení

Nový model s úplnou informací o stavu popisující trajektorii systému je definován v (5.54). Aktivní detektor chyb a regulátor pro úlohu s úplnou informací o stavu odpovídá

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \bar{\rho}^r(\boldsymbol{\xi}_k) = \begin{bmatrix} \bar{\sigma}^r(\boldsymbol{\xi}_k) \\ \bar{\gamma}^r(\boldsymbol{\xi}_k) \end{bmatrix}, \quad (5.59)$$

kde $\bar{\rho}^r : \mathcal{G} \mapsto \mathcal{M} \times \mathcal{U}$, $\bar{\sigma}^r : \mathcal{G} \mapsto \mathcal{M}$ a $\bar{\gamma}^r : \mathcal{G} \mapsto \mathcal{U}$ jsou hledané strategie. Původní kritérium (4.19) je možné po přeformulování zapsat jako

$$\bar{J} = \lim_{F \rightarrow \infty} \mathbb{E} \left\{ \sum_{k=0}^F \eta^k \bar{L}(\boldsymbol{\xi}_k, d_k, \mathbf{u}_k) \right\}, \quad (5.60)$$

kde $\bar{L} : \mathcal{G} \times \mathcal{M} \times \mathcal{U} \mapsto \mathbb{R}^+$ je funkce ekvivalentní původní ztrátové funkci a strukturou podobnou té v (5.50), neboli

$$\bar{L}(\boldsymbol{\xi}_k, d_k, \mathbf{u}_k) = \alpha \bar{L}^d(\boldsymbol{\xi}_k, d_k) + (1 - \alpha) \bar{L}^c(\boldsymbol{\xi}_k, \mathbf{u}_k). \quad (5.61)$$

Ztrátové funkce $\bar{L}^d : \mathcal{G} \times \mathcal{M} \mapsto \mathbb{R}^+$ a $\bar{L}^c : \mathcal{G} \times \mathcal{U} \mapsto \mathbb{R}^+$ aproximující původní ztrátové funkce L^d a L^c jsou definovány jako

$$\bar{L}^d(\boldsymbol{\xi}_k, d_k) \approx \mathbb{E} \left\{ L^d(\mu_k, d_k) | \mathbf{Z}_0^k, d_k \right\}, \quad (5.62)$$

$$\bar{L}^c(\boldsymbol{\xi}_k, \mathbf{u}_k) \approx \mathbb{E} \left\{ L^c(\mathbf{x}_k, \mathbf{u}_k) | \mathbf{Z}_0^k, \mathbf{u}_k \right\}. \quad (5.63)$$

V této kapitole byl představen možný postup transformace původně formulované úlohy s neúplnou informací o stavu na úlohu s úplnou informací o stavu a celý postup přeformulování byl ilustrován na obrázku 5.2. Postup hledání řešení úlohy s úplnou informací o stavu je diskutován v následující kapitole 6.

Kapitola 6

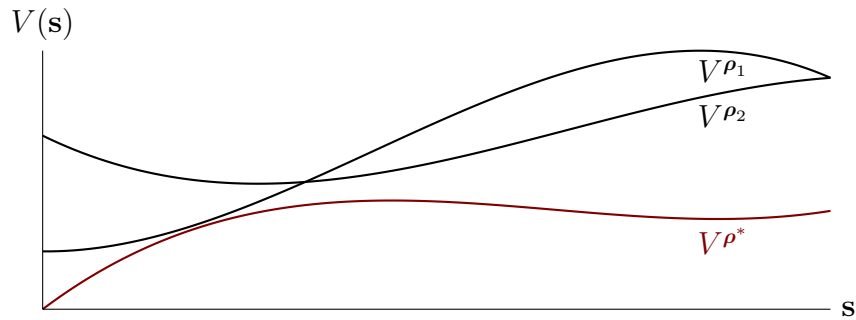
Řešení optimalizační úlohy s úplnou informací o stavu

Tato kapitola se zabývá řešením optimalizační úlohy aktivní detekce chyb na nekonečném časovém horizontu s úplnou informací o stavu. První podkapitola 6.1 je věnována obecnému úvodu k dynamickému programování a propojení s úlohou návrhu aktivního detektoru chyb. Z důvodů snížení výpočetní složitosti je nutné navrhnout aproximační postupy získání suboptimálního řešení, kterým je věnována převážná část dalšího textu. Podkapitoly 6.2 a 6.3 se zabývají aproximačními postupy založenými na kvantizaci stavového prostoru, podkapitoly 6.4 a 6.5 jsou založené na aproximativních spojitých funkcích. V podkapitole 6.6 je na závěr souhrnně diskutováno řešení úlohy aktivní detekce chyb a řízení.

6.1 Dynamické programování

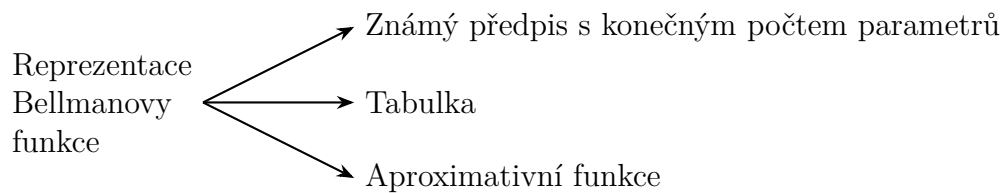
Dynamické programování je technika, kterou lze použít pro nalezení optimální strategie při znalosti úplného popisu systému jako Markovova rozhodovacího procesu [93]. Princip dynamického programování spočívá v transformaci složité úlohy na posloupnosti jednodušších úloh. Tato dekompozice vede k zavedení *Bellmanovy funkce*, která se v kontextu zpětnovazebního učení nazývá také jako *užitková funkce*. Bellmanova funkce popisuje očekávané dlouhodobé ztráty při použití určité strategie. Návrh aktivního detektoru chyb je tedy založen na nalezení Bellmanovy funkce.

Jedním z poznatků dynamického programování je existence alespoň jedné strategie, pro kterou jsou hodnoty Bellmanovy funkce pro všechny stavy minimální. Existence optimální strategie je ilustrováno na obrázku 6.1 a důkaz pro diskrétní stavy a diskrétní vstupy je uveden v [94]. V důsledku existuje právě jedna optimální Bellmanova funkce. Optimální Bellmanova funkce může být získána řešením Bellmanovy funkcionální rovnice [50]. Způsob hledání řešení Bellmanovy funkcionální rovnice a samotná reprezentace Bellmanovy funkce záleží na konkrétní úloze. Různé druhy reprezentace Bellmanovy funkce jsou uvedeny na obrázku 6.2. Analytické řešení Bellmanovy funkcionální rovnice existuje ve speciálních případech jako je optimalizační úloha lineárního kvadraticky optimálního



Obrázek 6.1: Ilustrace existence alespoň jedné strategie ρ^* pro kterou platí $V^{\rho^*}(\mathbf{s}) \leq V^{\rho}(\mathbf{s})$ pro všechny stavy \mathbf{s} .

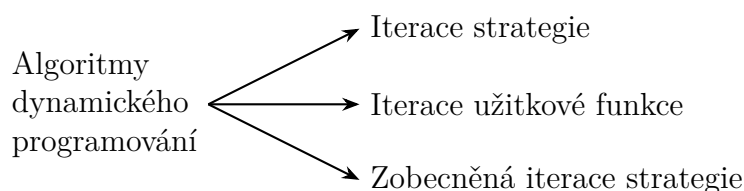
řízení (LQR). Bellmanova funkce je pak reprezentována známým funkčním předpisem s konečným počtem parametrů. V případě LQR se jedná o kvadratickou formu, kdy neznámé parametry jsou dány řešením algebraické Riccatiho rovnice [79]. Dalším faktorem ovlivňujícím způsob hledání řešení Bellmanovy funkcionální rovnice a reprezentaci Bellmanovy funkce je celkový počet stavů. V případě diskrétního stavu s konečným počtem prvků lze Bellmanovu funkci reprezentovat tabulkou. V obecném případě spojitého stavu nebo velmi rozměrného diskrétního stavového prostoru je pak nutné použít nějakou aproximativní reprezentaci Bellmanovy funkce [95].



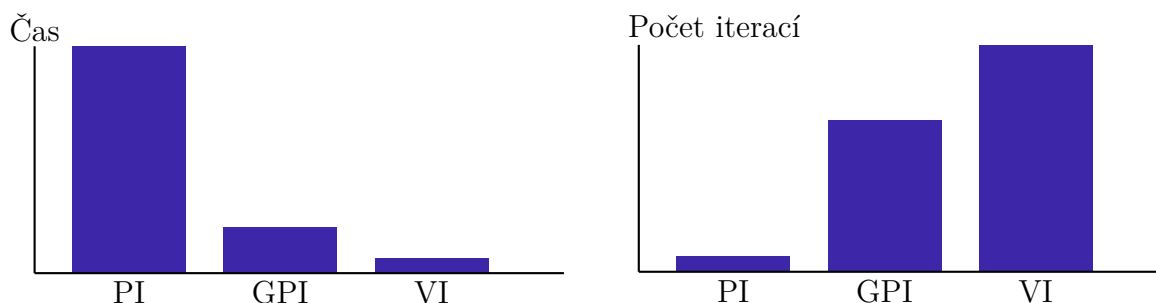
Obrázek 6.2: Přehled základních reprezentací Bellmanovy funkce.

V obecné optimalizační úloze diskrétního nebo spojitého prostoru stavů může být řešení úlohy na nekonečném časovém horizontu nalezeno pomocí iteračních algoritmů dynamického programování, jejichž rozdělení je ilustrováno na obrázku 6.3. Jedná se o iteraci strategie, zobecněnou iteraci strategie a iteraci užtkové funkce [Š10], [96]. Uvedené algoritmy vyžadují znalost modelu systému a tvoří je dva kroky, které se opakují do splnění předem definovaných ukončovacích podmínek. V každé iteraci je nejprve ohodnocena aktuální strategie nalezením příslušné Bellmanovy funkce na základě funkcionální rovnice. V následujícím kroku je aktuální strategie zlepšena na základě nalezené Bellmanovy funkce. Stěžejním rozdílem algoritmů je způsob nalezení Bellmanovy funkce aktuální strategie v kroku ohodnocení strategie. Tento krok spočívá v řešení funkcionální rovnice, která pro konečný počet stavů nebo konečný počet parametrů aproximativní reprezentace Bellmanovy funkce tvoří soustavu lineárních rovnic. Algoritmus iterace strategie nalezne řešení neiterační metodou jako je například Gaussova eliminační metoda. Takový postup má kubickou složitost. Z důvodu snížení výpočetní náročnosti tohoto kroku je možné využít numerickou metodu

postupné aproximace, která spočívá ve vytvoření posloupnosti odhadů Bellmanovy funkce konvergujících k jejímu skutečnému tvaru. V případě zobecněné iterace strategie je aproximace Bellmanovy funkce aktuální strategie nalezena několika vnitřními iteracemi postupné aproximace, zatímco iterace užitkové funkce využívá právě jednu vnitřní iteraci metody postupné aproximace. Ilustrace výpočetní složitosti diskutovaných algoritmů je ukázána na obrázku 6.4. V důsledku rozdílného způsobu řešení je časová výpočetní náročnost kroku ohodnocení strategie nejvyšší pro algoritmus iterace strategie a nejnižší pro algoritmus iterace užitkové funkce. Naopak aproximační ohodnocení strategie zpomalí celkovou konvergenci algoritmů zobecněné iterace strategie a iterace užitkové funkce. Pro konečný počet stavů iterace strategie konverguje v konečném počtu iterací, zatímco druhé dva algoritmy konvergují asymptoticky [79]. Z toho důvodu se obvykle volí ukončovací podmínka, která zajistí ukončení iteračního algoritmu po dosažení konečného počtu kroků. Uvedené poměry výpočetní složitosti z obrázku 6.4 jsou pouze ilustrativní.



Obrázek 6.3: Přehled algoritmů dynamického programování.



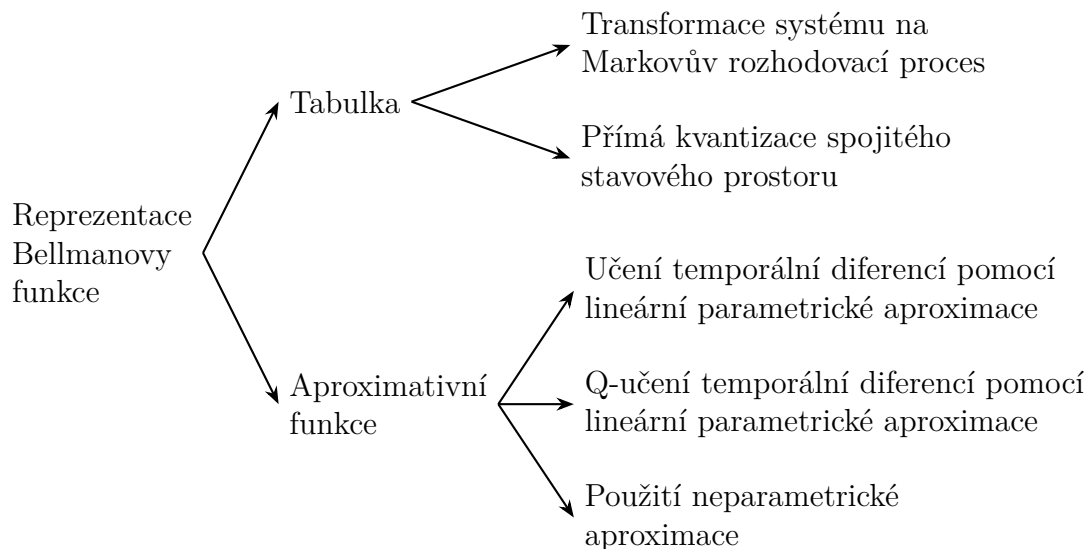
(a) Časová výpočetní náročnosti kroku ohodnocení strategie.

(b) Počet iterací potřebných k nalezení strategie vyhovující ukončovací podmínce.

Obrázek 6.4: Ilustrace výpočetní složitosti iterace strategie (PI), zobecněné iterace strategie (GPI) a iterace užitkové funkce (VI).

Jak bylo naznačeno v předchozím odstavci, řešení Bellmanovy funkcionální rovnice může být pro konečný diskrétní prostor stavů nalezeno iteračními algoritmy dynamického programování. V případě optimalizační úlohy aktivní detekce chyb je množina stavů, přesněji hyperstavů, spojitá, a proto je řešení podobným způsobem výpočetně a paměťově nereálné. Tento fenomén bývá nazýván jako prokletí rozměrnosti [97]. *Prokletí rozměrnosti* v podstatě představuje exponenciální růst výpočetní složitosti a paměťových požadavků

s rostoucím počtem možných stavů systému. Prokletí rozměrnosti slouží jako motivace pro řešení optimalizačních úloh pomocí různých aproximačních postupů. V této práci je navrženo několik aproximačních postupů pro aktivní detekci chyb. Jejich rozdělení v závislosti na reprezentaci Bellmanovy funkce je ilustrováno na obrázku 6.5. Vhodný výběr aproximace Bellmanovy funkce má zásadní vliv na kvalitu navrženého aktivního detektoru chyb. Důležitou roli zároveň hraje využití apriorní znalosti o úloze. První dvě techniky využívají pro reprezentaci Bellmanovy funkce tabulku. V prvním případě je model systému aproximován Markovovým rozhodovacím procesem. Markovův rozhodovací proces [98], [99] je definován pomocí uspořádané čtveřice: diskrétní prostor stavů, diskrétní prostor vstupů, matice pravděpodobností přechodů a ztrátová funkce. V druhém případě se využívá agregační funkce, která transformuje spojitý stav na nový stav z diskrétní množiny, která je vhodně zvolena tak, aby pokrývala podstatnou část původního spojitého stavového prostoru. Tyto techniky lze použít především v případě méně rozměrného stavu, a proto jsou v této práci prezentovány pro hyperstav ζ_k definovaný v (5.41). Druhá navržená skupina technik využívá lineární aproximaci Bellmanovy funkce, která může být neparametrická nebo parametrická. V případě parametrické struktury jsou v práci diskutovány dvě techniky založené na učení temporální diferencí, která představuje jednu ze základních technik zpětnovazebního učení [93]. Tyto techniky jsou prezentované pro obecnější strukturu informačních statistik, tedy pro hyperstav ξ_k definovaný v (5.53).



Obrázek 6.5: Přehled vybraných aproximačních postupů řešení úlohy návrhu aktivního detektoru chyb.

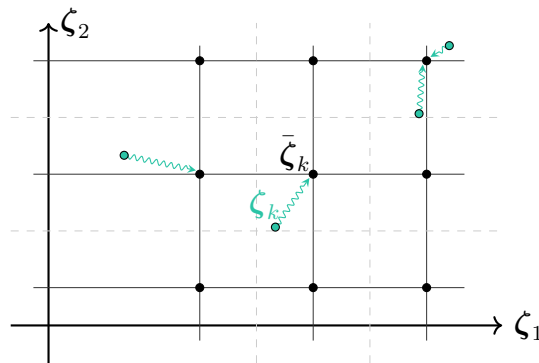
Základním stavebním kamenem dynamického programování je vlastnost kontrakce, na základě které lze dokázat konvergenci iteračních algoritmů dynamického programování [95]. Avšak při zavedení aproximace Bellmanovy funkce může být vlastnost kontrakce porušena, a tím narušena samotná konvergence iteračních algoritmů. Proto je nutné při aproximačním řešení optimalizační úlohy postupovat obezřetně.

6.2 Transformace modelu na Markovův rozhodovací proces

Mnoho dynamických systémů může být modelováno jako Markovovy rozhodovací procesy [98], [99]. Za předpokladu dostatečně malého konečného počtu stavů a vstupů může být řešení optimalizační úlohy pro Markovovy rozhodovací procesy na nekonečném horizontu nalezeno pomocí iteračních algoritmů dynamického programování, jak bylo naznačeno v předchozí podkapitole. S ohledem na současný výpočetní výkon a paměťové možnosti lze za dostatečně malý počet stavů považovat stovky tisíc až miliony v případě řídké matice pravděpodobnosti přechodů. V první části této podkapitoly je prezentována možná transformace modelu na Markovův rozhodovací proces. Druhá část se zabývá samotným řešením optimalizační úlohy návrhu aktivního detektoru chyb pomocí dynamického programování.

6.2.1 Přeformulování úlohy

Nechť je Markovův rozhodovací proces definován pomocí uspořádané čtveřice $(\mathcal{R}^g, \mathcal{U}, \mathbf{P}, \bar{L}^d)$, kde \mathcal{R}^g je konečný diskretní prostor hyperstavů $\bar{\zeta}_k \in \mathcal{R}^g$, \mathcal{U} je konečná diskretní množina vstupů $\mathbf{u}_k \in \mathcal{U}$, \mathbf{P} je matice podmíněných pravděpodobností přechodů $P_{i,j,l} = P(\bar{\zeta}_{k+1} = \bar{\zeta}^j | \bar{\zeta}_k = \bar{\zeta}^i, \mathbf{u}_k = \bar{\mathbf{u}}^l)$ a $\bar{L}^d : \mathcal{R}^g \times \mathcal{M} \mapsto \mathbb{R}$ je ztrátová funkce detekce chyb. Transformace systému v zásadě spočívá ve vytvoření diskretní množiny \mathcal{R}^g tak, aby dostatečně pokryla všechny důležité regiony spojitého prostoru hyperstavů \mathcal{R} . Promítnutí spojitého hyperstavu $\zeta \in \mathcal{R}$ na diskretní množinu \mathcal{R}^g je provedeno pomocí tzv. agregační funkce. Podmíněné pravděpodobnosti přechodů mohou být vypočítány různými technikami. Jako příklad lze uvést metodu Monte Carlo (MC), kdy je matice podmíněných pravděpodobností přechodů \mathbf{P} vytvořena na základě dostatečného množství simulací trajektorie systému a agregace hyperstavu. Vysoké výpočetní nároky této metody nemusí být zásadní překážkou, jelikož výpočet podmíněných pravděpodobností přechodů je pro daný model proveden pouze jednou a off-line.



Obrázek 6.6: Ilustrace průmětu spojitého hyperstavu ζ_k a dalších spojitých hyperstavů do diskretního prostoru hyperstavů pomocí agregační funkce ϑ .

Nechť je spojitý prostor hyperstavů \mathcal{R} kvantizován mřížkou tak, aby odpovídající diskrétní prostor hyperstavů byl definován jako kartézský součin

$$\mathcal{R}^g \equiv \mathcal{R}_1^g \times \mathcal{R}_2^g \times \dots \times \mathcal{R}_{n_\zeta}^g = \left\{ \bar{\zeta}^1, \bar{\zeta}^2, \dots, \bar{\zeta}^{N_{\bar{\zeta}}} \right\}, \quad (6.1)$$

kde $\bar{\zeta}^t \in \mathbb{R}^{n_\zeta}$, $t \in \{1, 2, \dots, N_{\bar{\zeta}}\}$ je diskrétní hyperstav. Diskrétní množiny $\mathcal{R}_j^g \subset \mathbb{R}$, $j = 1, 2, \dots, n_\zeta$, jsou vybrány na základě apriorní informace tak, aby byly konečné a zároveň pokrývaly všechny důležité oblasti původního spojitého prostoru \mathcal{R} , kde pravděpodobnost výskytu hyperstavu je nezanedbatelná. Každý spojitý hyperstav ζ_k může být promítnut do prostoru \mathcal{R}^g pomocí následující zvolené agregační funkce

$$\bar{\zeta}_k = \vartheta(\zeta_k) = \arg \min_{\mathbf{z} \in \mathcal{R}^g} \|\zeta_k - \mathbf{z}\|_2, \quad (6.2)$$

kde $\vartheta : \mathcal{R} \mapsto \mathcal{R}^g$ je agregační funkce a \mathbf{z} je prvek z diskrétní množiny hyperstavů \mathcal{R}^g . Průmět spojitých hyperstavů do diskrétního prostoru hyperstavů využívající agregační funkci ϑ definovanou v (6.2) je ilustrován na obrázku 6.6. Celkový počet diskrétních hyperstavů roste exponenciálně s dimenzí hyperstavu.

Nechť $\mathcal{O}(\bar{\zeta}^j)$ je okolí diskrétního hyperstavu $\bar{\zeta}^j \in \mathcal{R}^g$ definováno pomocí agregační funkce ϑ následovně

$$\mathcal{O}(\bar{\zeta}^j) = \{\zeta \in \mathcal{R} : \vartheta(\zeta) = \bar{\zeta}^j\}, \quad (6.3)$$

neboli okolí hyperstavu $\bar{\zeta}^j$ je tvořeno všemi hyperstavy $\zeta \in \mathcal{R}$, které se skrz agregační funkci ϑ promítnou právě do hyperstavu $\bar{\zeta}^j$. Poté podmíněné pravděpodobnosti přechodů mezi hyperstavy mohou být definovány jako [100]

$$\begin{aligned} P_{i,j,l} &= P(\zeta_{k+1} \in \mathcal{O}(\bar{\zeta}^j) | \zeta_k = \bar{\zeta}^i, \mathbf{u}_k = \bar{\mathbf{u}}^l) \\ &= \int_{\mathcal{O}(\bar{\zeta}^j)} p(\zeta_{k+1} | \zeta_k = \bar{\zeta}^i, \mathbf{u}_k = \bar{\mathbf{u}}^l) d\zeta_{k+1}. \end{aligned} \quad (6.4)$$

Podmíněná pravděpodobnost přechodu $P_{i,j,l}$ odpovídá pravděpodobnosti, že diskrétní hyperstav $\bar{\zeta}^i$ za použití vstupu $\bar{\mathbf{u}}^l$ přejde do okolí hyperstavu $\bar{\zeta}^j$. Dále $P_{i,j,l}$ musí splňovat následující podmínku

$$P_{i,j,l} \geq 0 \text{ pro všechna } i, j, l \text{ a zároveň } \sum_{j=1}^{N_{\bar{\zeta}}} P_{i,j,l} = 1 \text{ pro všechna } i, l. \quad (6.5)$$

Diskrétní prostor hyperstavů spolu s maticí podmíněných pravděpodobností přechodů zásadně ovlivňují přesnost aproximace, a proto je důležité věnovat jejich určení zvýšenou pozornost.

Původní model systému s úplnou informací o stavu (5.43) může být chápán jako Markovův rozhodovací proces definovaný pomocí čtveřice $(\mathcal{R}^g, \mathcal{U}, \mathbf{P}, \bar{L}^d)$. Aktivní detektor chyb je v případě zvolené transformace reprezentován následovně

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \bar{\rho}^q(\bar{\zeta}_k) = \begin{bmatrix} \bar{\sigma}^q(\bar{\zeta}_k) \\ \bar{\gamma}^q(\bar{\zeta}_k) \end{bmatrix}, \quad (6.6)$$

kde $\bar{\rho}^q : \mathcal{R}^g \mapsto \mathcal{M} \times \mathcal{U}$, $\bar{\sigma}^q : \mathcal{R}^g \mapsto \mathcal{M}$ a $\bar{\gamma}^q : \mathcal{R}^g \mapsto \mathcal{U}$ jsou hledané strategie. Cílem optimalizace je minimalizovat diskontované aditivní kritérium kvality detekce chyb

$$J = \lim_{F \rightarrow \infty} \mathbb{E} \left\{ \sum_{k=0}^F \eta^k \bar{L}^d(\bar{\zeta}_k, d_k) \right\}, \quad (6.7)$$

kde diskontní faktor $\eta \in (0, 1)$ umožňuje s rostoucím časem snižovat význam budoucích ztrát. Takto přeformulovaná úloha může být vyřešena pomocí dynamického programování.

6.2.2 Řešení dynamickým programováním

Bellmanova funkce $V_{\bar{\rho}^q} : \mathcal{R}^g \mapsto \mathbb{R}$, udávající celkové budoucí ztráty z hyperstavu $\bar{\zeta}_k$ použitím strategie $\bar{\rho}^q$, může být definována jako

$$V_{\bar{\rho}^q}(\bar{\zeta}_k) = \lim_{F \rightarrow \infty} \mathbb{E} \left\{ \sum_{i=k}^F \eta^i \bar{L}^d(\bar{\zeta}_i, \bar{\sigma}^q(\bar{\zeta}_i)) | \bar{\zeta}_k \right\}. \quad (6.8)$$

Pro jednoduchost bude v následující části vynechán časový index. Na základě Bellmanova principu optimality [49], [84] splňuje Bellmanova funkce $V_{\bar{\rho}^q}$ v případě Markovových rozhodovacích úloh následující funkcionální rovnici

$$V_{\bar{\rho}^q}(\bar{\zeta}^i) = \bar{L}^d(\bar{\zeta}^i, \bar{\sigma}^q(\bar{\zeta}^i)) + \eta \sum_{j=1}^{N_{\bar{\zeta}}} P_{i,j}^{\bar{\gamma}^q} V_{\bar{\rho}^q}(\bar{\zeta}^j), \quad i = 1, 2, \dots, N_{\bar{\zeta}}, \quad (6.9)$$

kde pravděpodobnosti přechodů $P_{i,j}^{\bar{\gamma}^q}$ jsou definovány jako

$$P_{i,j}^{\bar{\gamma}^q} = P(\bar{\zeta}_{k+1} = \bar{\zeta}^j | \bar{\zeta}_k = \bar{\zeta}^i, \mathbf{u}_k = \bar{\gamma}^q(\bar{\zeta}^i)). \quad (6.10)$$

Optimální Bellmanova funkce $V^{q*} : \mathcal{R}^g \mapsto \mathbb{R}$ pro danou úlohu je definovaná jako

$$V^{q*}(\bar{\zeta}^i) = \min_{\bar{\rho}^q} V_{\bar{\rho}^q}(\bar{\zeta}^i) \quad (6.11)$$

pro všechny možné hyperstavy $\bar{\zeta}^i \in \mathcal{R}^g$ a vyhovuje následující Bellmanově funkcionální rovnici

$$V^{q*}(\bar{\zeta}^i) = \min_{d \in \mathcal{M}, \mathbf{u} \in \mathcal{U}} \left\{ \bar{L}^d(\bar{\zeta}^i, d) + \eta \sum_{j=1}^{N_{\bar{\zeta}}} P_{i,j}^{\mathbf{u}} V^{q*}(\bar{\zeta}^j) \right\}, \quad (6.12)$$

kde $P_{i,j}^{\mathbf{u}} = P(\bar{\zeta}_{k+1} = \bar{\zeta}^j | \bar{\zeta}_k = \bar{\zeta}^i, \mathbf{u}_k = \mathbf{u})$. V tomto případě je z důvodu zjednodušení zápisu nesprávným způsobem použita konvence značení u pravděpodobností přechodů $P_{i,j}^{\mathbf{u}}$, jelikož v dříve uvedené definici (6.10) je místo hodnoty v horním index použita funkce. Stojí za zmínku, že Bellmanova funkce V^{q*} je optimální pro Markovův rozhodovací proces,

avšak pro celkové řešení optimalizační úlohy návrhu aktivního detektoru chyb je suboptimální z důvodu aproximace původního modelu systému. Jelikož ztrátová funkce \bar{L}^d v případě úlohy aktivní detekce chyb nezávisí na vstupu \mathbf{u} a trajektorie hyperstavu naopak nezávisí na generovaných rozhodnutích d , může být Bellmanova funkcionální rovnice (6.12) přepsána do tvaru

$$V^{q^*}(\bar{\zeta}^i) = \min_{d \in \mathcal{M}} \{ \bar{L}^d(\bar{\zeta}^i, d) \} + \eta \min_{\mathbf{u} \in \mathcal{U}} \left\{ \sum_{j=1}^{N_{\bar{\zeta}}} P_{i,j}^{\mathbf{u}} V^{q^*}(\bar{\zeta}^j) \right\}. \quad (6.13)$$

Rozhodnutí generované optimálním generátorem rozhodnutí minimalizuje aktuální ztráty a vstup z optimálního generátoru vstupního signálu minimalizuje očekávané budoucí ztráty. Aktivní detektor chyb $\bar{\rho}^{q^*} = [\bar{\sigma}^{q^*}, (\bar{\gamma}^{q^*})^T]^T$ pak může být vyjádřen jako

$$\bar{\sigma}^{q^*}(\bar{\zeta}) = \arg \min_{d \in \mathcal{M}} \bar{L}^d(\bar{\zeta}, d), \quad (6.14)$$

$$\bar{\gamma}^{q^*}(\bar{\zeta}) = \arg \min_{\mathbf{u} \in \mathcal{U}} \left\{ \sum_{j=1}^{N_{\bar{\zeta}}} P_{i,j}^{\mathbf{u}} V^{q^*}(\bar{\zeta}^j) \right\}, \quad (6.15)$$

kde $\bar{\rho}^{q^*} : \mathcal{R}^g \mapsto \mathcal{M} \times \mathcal{U}$, $\bar{\sigma}^{q^*} : \mathcal{R}^g \mapsto \mathcal{M}$ a $\bar{\gamma}^{q^*} : \mathcal{R}^g \mapsto \mathcal{U}$ jsou příslušné strategie. Z výrazu (6.15) byl odstraněn diskontní faktor vzhledem k vlastnostem funkce $\arg \min$. Řešení funkcionální rovnice (6.13) se využije pro generování budícího vstupního signálu pomocí strategie $\bar{\gamma}^{q^*}$. Závislost strategie $\bar{\gamma}^{q^*}$ na Bellmanově funkci V^{q^*} je zřejmá ze vztahu (6.15).

Řešení Bellmanovy funkcionální rovnice (6.13) může být nalezeno iteračním postupem, který je tvořen neustálým opakováním dvou kroků: ohodnocení strategie $\bar{\rho}^{q^{(i)}}$ a vylepšení strategie. V kroku *ohodnocení strategie* je Bellmanova funkce $V_{\bar{\rho}^{q^{(i)}}}^{q^{(i)}}$ nalezena řešením rovnice (6.9), která může být zapsána jako systém lineárních rovnic

$$(\mathbf{I}_{N_{\bar{\zeta}}} - \eta \mathbf{P}^{\bar{\gamma}^{q^{(i)}}}) \mathbf{v}^{\bar{\rho}^{q^{(i)}}} = \mathbf{l}^{\bar{\rho}^{q^{(i)}}}, \quad (6.16)$$

kde $i = 0, 1, 2, \dots$ je iterační index algoritmu, $\mathbf{P}^{\bar{\gamma}^{q^{(i)}}}$ je $N_{\bar{\zeta}} \times N_{\bar{\zeta}}$ známá matice pravděpodobností přechodu při použití strategie $\bar{\rho}^{q^{(i)}}$, $\mathbf{I}_{N_{\bar{\zeta}}}$ je jednotková matice rozměru $N_{\bar{\zeta}} \times N_{\bar{\zeta}}$, $\mathbf{l}^{\bar{\rho}^{q^{(i)}}}$ je známý vektor s prvky $l_m^{\bar{\rho}^{q^{(i)}}} = \bar{L}^d(\bar{\zeta}^m, \bar{\sigma}^{q^{(i)}}(\bar{\zeta}^m))$ a $\mathbf{v}^{\bar{\rho}^{q^{(i)}}} \in \mathbb{R}^{N_{\bar{\zeta}}}$ je neznámý vektor, jehož prvky jsou definovány jako $v_m^{\bar{\rho}^{q^{(i)}}} = V_{\bar{\rho}^{q^{(i)}}}^{q^{(i)}}(\bar{\zeta}^m)$, $m = 1, 2, \dots, N_{\bar{\zeta}}$. V kroku *vylepšení strategie* je nová strategie $\bar{\rho}^{q^{(i+1)}} = [\bar{\sigma}^{q^{(i+1)}}, \bar{\gamma}^{q^{(i+1)}}]^T$ vypočítána použitím následující rovnice

$$\bar{\sigma}^{q^{(i+1)}}(\bar{\zeta}^m) = \arg \min_{d \in \mathcal{M}} \bar{L}^d(\bar{\zeta}^m, d), \quad (6.17)$$

$$\bar{\gamma}^{q^{(i+1)}}(\bar{\zeta}^m) = \arg \min_{\mathbf{u} \in \mathcal{U}} \left\{ \sum_{n=1}^{N_{\bar{\zeta}}} P_{m,n}^{\mathbf{u}} V_{\bar{\rho}^{q^{(i)}}}^{q^{(i)}}(\bar{\zeta}^n) \right\} \quad (6.18)$$

pro všechna $m = 1, 2, \dots, N_{\bar{\zeta}}$. Vztahy (6.17) a (6.18) jsou iteračním řešením (6.14) a (6.15). Ze vztahu (6.17) je patrné, že se strategie $\bar{\sigma}^{q^{(i)}}$ nemění s iteracemi a její výpočet může být

proveden na začátku algoritmu a pouze jednou. Z toho důvodu lze aktivní detektor chyb v každém iteračním kroku i definovat jako

$$\bar{\rho}^{q(i)}(\bar{\zeta}^m) = \begin{bmatrix} \bar{\sigma}^q(\bar{\zeta}^m) \\ \bar{\gamma}^{q(i)}(\bar{\zeta}^m) \end{bmatrix}, \quad (6.19)$$

kde $\bar{\sigma}^q = \bar{\sigma}^{q(0)}$. Kroky ohodnocení strategie a vylepšení strategie jsou s přibývajícím iteracemi opakovány, dokud není splněna ukončovací podmínka celého iteračního algoritmu. Velmi často se využívá porovnání nekonečné normy rozdílu mezi dvěma po sobě jdoucími Bellmanovými funkcemi s předem zvolenou mezí $\epsilon_V > 0$, $\|V_{\bar{\rho}^{q(i+1)}}(\bar{\zeta}) - V_{\bar{\rho}^{q(i)}}(\bar{\zeta})\|_\infty \leq \epsilon_V$, splnění požadavku na minimální počet iterací $i \geq i_{\min}$, nebo rovnosti dvou po sobě jdoucích strategií $\bar{\rho}^{q(i+1)} = \bar{\rho}^{q(i)}$.

Na závěr je vhodné uvést provázání této části s iteračními algoritmy dynamického programování, které byly uvedeny v podkapitole 6.1. Soustava lineárních rovnic (6.16) může být vyřešena iteračně. Iterace postupné aproximace je dána vztahem

$$\mathbf{v}_{(j+1)}^{\bar{\rho}^{q(i)}} = \mathbf{l}^{\bar{\rho}^{q(i)}} + \eta \mathbf{P}^{\bar{\gamma}^{q(i)}} \mathbf{v}_{(j)}^{\bar{\rho}^{q(i)}}, \quad (6.20)$$

kde $j = 0, 1, 2, \dots, \bar{j}$ je iterační index postupné aproximace a \bar{j} je parametr, který může být chápán jako nepřímý indikátor počtu postupných aproximací. Přestože počáteční vektor $\mathbf{v}_{(0)}^{\bar{\rho}^{q(i)}}$ může být libovolný, běžně je pro počáteční iteraci algoritmu zvolen nulový $\mathbf{v}_{(0)}^{\bar{\rho}^{q(0)}} = \mathbf{0}$ a dále jako $\mathbf{v}_{(0)}^{\bar{\rho}^{q(i)}} = \mathbf{v}_{(0)}^{\bar{\rho}^{q(i-1)}}$ pro $i = 1, 2, \dots$. V závislosti na hodnotě parametru \bar{j} lze definovat tři iterační algoritmy [96].

- V případě volby parametru $\bar{j} = 0$ je provedena pouze jedna iterace postupné aproximace soustavy lineárních rovnic. Algoritmus opakující krok ohodnocení strategie a krok vylepšení strategie se pak nazývá *iterace uživatelské funkce*. Asymptotická výpočetní náročnost jedné iterace algoritmu iterace uživatelské funkce je $\mathcal{O}(N_\xi^2 N_u)$.
- Opačný případ nastane, když $\bar{j} \rightarrow \infty$, neboli soustava lineárních rovnic je vyřešena neiterativní metodou, například Gaussovou eliminační metodou. Algoritmus založený na neiteračním způsobu řešení je známý jako *iterace strategie*. Celková výpočetní náročnost jedné iterace algoritmu iterace strategie odpovídá $\mathcal{O}(N_\xi^3 + N_\xi^2 N_u)$.
- Poslední algoritmus *zobecněné iterace strategie* odpovídá volbě nenulové konečné hodnoty parametru \bar{j} . Tento algoritmus obvykle představuje kompromis mezi rychlostí konvergence a výpočetními nároky a je vhodný zejména pro úlohy s velmi rozměrným prostorem hyperstavů, jelikož v takovém případě může být neiterativní hledání řešení systému lineárních rovnic (6.16) neúnosně výpočetně náročné.

Mezi nevýhody zobecněné iterace strategie patří složitější implementace v porovnání s iterací uživatelské funkce nebo netriviální volba konečné hodnoty parametru \bar{j} . Následující způsoby volby parametru \bar{j} jsou představeny v [101].

- V každé i -té iteraci algoritmu se použije nenulová konečná hodnota \bar{j} , která je zvolena předem.
- Parametr \bar{j} se mění v závislosti na indexu i , například s rostoucím i se zvyšuje hodnota \bar{j} .
- Parametr \bar{j} se mění v závislosti na předem definovaném ukazateli chování, například v závislosti na $\|V_{\bar{\rho}^{q(i)}}^{(j+1)}(\bar{\zeta}) - V_{\bar{\rho}^{q(i)}}^{(j)}(\bar{\zeta})\|_\infty$.

Přestože obecné zásady pro volbu parametru \bar{j} jsou známé, konkrétní algoritmy se v literatuře nevyskytují. Úkolem následující části je navrhnout adaptivní algoritmus pro stanovení počtu iterací postupné aproximace v kroku ohodnocení strategie, který zrychlí konvergenci při požadavku na danou přesnost nalezeného řešení.

6.2.3 Adaptivní algoritmus zobecněné iterace strategie

V této části je navržen adaptivní algoritmus zobecněné iterace strategie založený na porovnávání přesnosti postupné aproximace s předem stanovenou mezí přesnosti. V kroku ohodnocení strategie je pomocí odvozeného vztahu vypočítána hodnota \bar{j} , na základě které je rozhodnuto, zda bude soustava lineárních rovnic (6.16) řešena iterativně, nebo neiterativně pomocí Gaussovy eliminační metody.

Algoritmus je odvozený za předpokladu, že podmíněné pravděpodobnosti přechodu jsou reprezentovány plnou maticí, nikoliv řídkou maticí, a zároveň k hledání řešení soustavy lineárních rovnic (6.16) je použita základní Gaussova eliminační metoda. Stojí za zmínku, že v případě řídké matice pravděpodobností přechodů se výpočetní nároky mohou volbou vhodných algoritmů zásadně snížit, a proto není zapotřebí tyto případy uvažovat zvlášť. Nejprůzřivější z hlediska výpočetních nároků pro velký počet hyperstavů $N_{\bar{\zeta}}$ je v případě zobecněné iterace strategie s adaptivním počtem iterací postupné aproximace spíše volba malých hodnot parametru \bar{j} [102]. Z hlediska výpočetních nároků je zapotřebí $N_{\bar{\zeta}}^3/3 + N_{\bar{\zeta}}^2 + N_{\bar{\zeta}}/3$ operací pro výpočet (6.16) pomocí Gaussovy eliminační metody. Na druhou stranu jedna iterace postupné aproximace představuje náročnost $N_{\bar{\zeta}}^2$ operací. Aby navržená adaptivní metoda byla efektivní, zdá se rozumné zvolit \bar{j} tak, aby platilo $\bar{j} \leq N_{\bar{\zeta}}/3$.

Přesnost postupné aproximace v kroku ohodnocení strategie je omezena následovně [103]

$$\|V_{\bar{\rho}^{q(i)}}^*(\bar{\zeta}_k) - V_{\bar{\rho}^{q(i)}}^{(\bar{j}+1)}(\bar{\zeta}_k)\|_\infty \leq \frac{2\eta^{\bar{j}+1}}{(1-\eta)} \|\bar{L}^d(\bar{\zeta}_k, \bar{\rho}^{q(i)}(\bar{\zeta}_k))\|_\infty, \quad (6.21)$$

kde $V_{\bar{\rho}^{q(i)}}^*$ je Bellmanova funkce vyhovující rovnici (6.16). Ze vztahu (6.21) je patrné, že se zmenšujícím se diskontním faktorem η se zmenšuje také počet iterací postupné aproximace vyjádřený prostřednictvím \bar{j} . Necht $\epsilon_{SA} > 0$ je mez přesnosti postupné aproximace, která ohraničuje pravou stranu (6.21), neboli

$$\frac{2\eta^{\bar{j}+1}}{(1-\eta)} \|\bar{L}^d(\bar{\zeta}_k, \bar{\rho}^{q(i)}(\bar{\zeta}_k))\|_\infty \leq \epsilon_{SA}. \quad (6.22)$$

Úpravou tohoto vztahu lze odvodit požadavky na počet iterací postupné aproximace tak, aby byla splněna požadovaná přesnost. Parametr \bar{j} tedy musí splňovat následující nerovnost

$$\bar{j} \geq \left\lceil \frac{\log \left(\frac{(1-\eta)\epsilon_{SA}}{2\|\bar{L}^d(\bar{\zeta}_k, \bar{\rho}^{q(i)}(\bar{\zeta}_k))\|_\infty} \right)}{\log \eta} - 1 \right\rceil, \quad (6.23)$$

kde $\lceil \cdot \rceil$ je operátor zaokrouhlení nahoru na nejbližší celé číslo. Z vlastnosti diskontního faktoru η je zřejmé, že $\log \eta < 0$. Znaménko čitatele pravé strany (6.23) závisí na hodnotách $\bar{L}^d(\bar{\zeta}_k, \bar{\rho}^{q(i)}(\bar{\zeta}_k))$, ϵ_{SA} a η . Jelikož se předpokládá, že ϵ_{SA} a η jsou pevně dané, hodnota čitatele na pravé straně roste se zvyšující se hodnotou $\|\bar{L}^d(\bar{\zeta}_k, \bar{\rho}^{q(i)}(\bar{\zeta}_k))\|_\infty$. V důsledku to znamená, že \bar{j} musí růst se zvyšující se hodnotou této normy. Naopak malá hodnota $\|\bar{L}^d(\bar{\zeta}_k, \bar{\rho}^{q(i)}(\bar{\zeta}_k))\|_\infty$ může v závislosti na volbě ϵ_{SA} a η způsobit negativní hodnotu \bar{j} . Možnou interpretací tohoto případu je nadbytečnost vylepšení strategie. Adaptivní postup zobecněné iterace strategie je shrnut v algoritmu 4.

Algoritmus 4 Algoritmus adaptivní zobecněné iterace strategie pro úlohu aktivní detekce chyb.

Inicializace Výběr přípustné strategie $\bar{\rho}^{q(0)}$ a meze přesnosti ϵ_{SA} . Nastavení iteračních indexů $i, j = 0$.

1. Generátor rozhodnutí Výpočet generátoru rozhodnutí $\bar{\sigma}^q$ pomocí (6.17).

2. Ohodnocení strategie

2.1) Výpočet počtu iterací v postupné aproximaci \bar{j} použitím (6.23).

2.2) **Pokud** $\bar{j} > N_{\bar{\zeta}}/3$, ohodnocení strategie $\bar{\rho}^{q(i)}$ ze vztahu (6.16) pomocí Gaussovy eliminační metody,

jinak nastavení $\mathbf{v}_{(0)}^{\bar{\rho}^{q(0)}} = \mathbf{0}$ nebo $\mathbf{v}_{(0)}^{\bar{\rho}^{q(i)}} = \mathbf{v}_{(0)}^{\bar{\rho}^{q(i-1)}}$ pro $i = 1, 2, \dots$. Ohodnocení $\bar{\rho}^{q(i)}$ postupnou aproximací (6.20), $j \leftarrow j + 1$, dokud $j = \bar{j}$.

Výstup: Bellmanova funkce $V_{\bar{\rho}^{q(i)}}$ definovaná pro všechny možné hyperstavy.

3. Vylepšení strategie Výpočet nové strategie $\bar{\gamma}^{q(i+1)}$ na základě $V_{\bar{\rho}^{q(i)}}$ pomocí (6.18).

Výstup: Strategie $\bar{\rho}^{q(i+1)}$ se strukturou (6.19).

Nastavení $i \leftarrow i + 1$ a **skok zpět na krok 2** dokud není splněna ukončovací podmínka.

6.3 Přímá kvantizace spojitého stavového prostoru

Pro velmi rozměrný diskrétní prostor hyperstavů \mathcal{R}^g je výpočet matice podmíněných pravděpodobností přechodů, definované v (6.4), výpočetně a paměťově náročný. Tato část shrnuje alternativní způsob řešení bez nutnosti přeformulování modelu systému na Markovův rozhodovací proces. Podobně jako v podkapitole 6.2 je spojitý prostor hyperstavů

kvantizován pomocí mřížky, avšak trajektorie hyperstavu je určena původním modelem systému (5.43) a agregační funkcí (6.2).

Bellmanova funkcionální rovnice v jejím obecném tvaru pro diskrétní stochastické systémy je definována jako

$$V^*(\zeta_k) = \min_{d_k \in \mathcal{M}, \mathbf{u}_k \in \mathcal{U}} \mathbb{E} \left\{ \bar{L}^d(\zeta_k, d_k) + \eta V^*(\zeta_{k+1}) | \zeta_k, d_k, \mathbf{u}_k \right\}, \quad (6.24)$$

kde $V^* : \mathcal{R} \mapsto \mathbb{R}$ je optimální Bellmanova funkce a $\zeta_{k+1} \in \mathcal{R}$ je hyperstav, do kterého se systém dostane z hyperstavu $\zeta_k \in \mathcal{R}$ použitím vstupu $\mathbf{u}_k \in \mathcal{U}$. Jelikož trajektorie hyperstavu nezávisí na rozhodnutí d_k a ztráty v případě aktivní detekce chyb nejsou ovlivněny vstupem \mathbf{u}_k , lze Bellmanovu funkcionální rovnici (6.24) upravit do následujícího tvaru

$$V^*(\zeta_k) = \min_{d_k \in \mathcal{M}} \mathbb{E} \left\{ \bar{L}^d(\zeta_k, d_k) | \zeta_k, d_k \right\} + \eta \min_{\mathbf{u}_k \in \mathcal{U}} \mathbb{E} \left\{ V^*(\zeta_{k+1}) | \zeta_k, \mathbf{u}_k \right\}. \quad (6.25)$$

Optimální generátor rozhodnutí $\bar{\sigma}^* : \mathcal{R} \mapsto \mathcal{M}$ a generátor vstupního signálu $\bar{\gamma}^* : \mathcal{R} \mapsto \mathcal{U}$ jsou dány následujícími vztahy

$$\bar{\sigma}^*(\zeta_k) = \arg \min_{d_k \in \mathcal{M}} \bar{L}^d(\zeta_k, d_k), \quad (6.26)$$

$$\bar{\gamma}^*(\zeta_k) = \arg \min_{\mathbf{u}_k \in \mathcal{U}} \mathbb{E} \left\{ V^*(\zeta_{k+1}) | \zeta_k, \mathbf{u}_k \right\}, \quad (6.27)$$

kde ve vztahu (6.26) mohla být z matematického výrazu díky deterministickým vlastnostem \bar{L}^d odstraněna střední hodnota. Zároveň stojí za povšimnutí, že z výrazu (6.27) byl odstraněn diskontní faktor vzhledem k vlastnostem funkce $\arg \min$. Ze vztahů (6.26) a (6.27) je patrné, že optimální generátor vstupního signálu $\bar{\gamma}^*$ je funkcí optimální Bellmanovy funkce V^* na rozdíl od optimálního generátoru rozhodnutí $\bar{\sigma}^*$.

Nechť je konečná diskrétní množina hyperstavů \mathcal{R}^g definována jako v (6.1) a agregační funkce ϑ zobrazující libovolný hyperstav $\zeta \in \mathcal{R}$ na odpovídající diskrétní hyperstav $\bar{\zeta} \in \mathcal{R}^g$ definována jako v (6.2). Úkolem je najít řešení Bellmanovy funkcionální rovnice (6.25), které představuje optimální Bellmanova funkce V^* . Jelikož zvolený způsob hledání řešení je pouze aproximativní, výsledné řešení je také aproximativní, přesněji řečeno výstupem je po částech konstantní funkce aproximující původní V^* pro všechny možné hyperstavy $\zeta \in \mathcal{R}$. Nalezená Bellmanova funkce se dále použije pro generování pomocného vstupního signálu pomocí (6.27). Jelikož prostor \mathcal{R}^g je diskrétní a konečný, lze k hledání Bellmanovy funkce opět použít algoritmy dynamického programování, například iteraci užitkové funkce nebo iteraci strategie.

V každé iteraci $i = 0, 1, 2, \dots$ algoritmu iterace užitkové funkce je vylepšena Bellmanova funkce aktuální strategie. Jelikož počet iterací postupně aproximace v kroku ohodnocení strategie je roven jedné, lze krok ohodnocení strategie a krok vylepšení strategie sloučit do jednoho ekvivalentního kroku [79], který má v případě formulované úlohy aktivní detekce chyb formu

$$\bar{V}^{(i+1)}(\bar{\zeta}) = \min_{d \in \mathcal{M}} \bar{L}^d(\bar{\zeta}, d) + \eta \min_{\mathbf{u} \in \mathcal{U}} \mathbb{E} \left\{ \bar{V}^{(i)}(\bar{\zeta}') | \bar{\zeta}, \mathbf{u} \right\}, \quad (6.28)$$

kde $\bar{V}^{(i)} : \mathcal{R}^g \mapsto \mathbb{R}$ je aproximace Bellmanovy funkce v i -té iteraci a $\bar{\zeta}' \in \mathcal{R}^g$ je diskrétní hyperstav, který je získán pomocí nelineární funkce

$$\bar{\zeta}' = \boldsymbol{\vartheta}(\boldsymbol{\phi}(\bar{\zeta}, \mathbf{u}_k, \mathbf{x}')), \quad (6.29)$$

kde stav \mathbf{x}' odpovídá \mathbf{x}_{k+1} . Počáteční podmínka $V_{\bar{\rho}^{(0)}}$ může být libovolná, běžně se však používá volba nulové Bellmanovy funkce.

Další možností hledání řešení rovnice (6.25) je použití algoritmu iterace strategie. V kroku ohodnocení strategie je nalezena aproximace Bellmanovy funkce $\bar{V}_{\bar{\rho}^{(i)}}$ odpovídající strategii $\bar{\rho}^{(i)}$ řešením následující funkcionální rovnice

$$\bar{V}_{\bar{\rho}^{(i)}}(\bar{\zeta}) = \bar{L}^d(\bar{\zeta}, \bar{\sigma}^q) + \eta \mathbb{E} \left\{ \bar{V}_{\bar{\rho}^{(i)}}(\bar{\zeta}') | \bar{\zeta}, \bar{\rho}^{(i)} \right\}, \quad (6.30)$$

kde $\bar{\sigma}^q$ je generátor rozhodnutí, který může být vypočítán jednorázově na začátku algoritmu pro všechny $\bar{\zeta} \in \mathcal{R}^g$ jako

$$\bar{\sigma}^q(\bar{\zeta}) = \arg \min_{d \in \mathcal{M}} \bar{L}^d(\bar{\zeta}, d). \quad (6.31)$$

Řešení funkcionální rovnice (6.30) je v případě algoritmu iterace strategie stanoveno řešením soustavy lineárních rovnic. V kroku vylepšení strategie je nalezeno řešení optimalizační úlohy

$$\bar{\gamma}^{q(i+1)}(\bar{\zeta}) = \arg \min_{\mathbf{u} \in \mathcal{U}} \mathbb{E} \left\{ \bar{V}_{\bar{\rho}^{(i)}}(\bar{\zeta}') | \bar{\zeta}, \mathbf{u} \right\}. \quad (6.32)$$

V každé iteraci i je dočasná strategie reprezentující aktivní detektor chyb definována $\bar{\rho}^{(i)} = [\bar{\sigma}^q, (\bar{\gamma}^{q(i)})^T]^T$. Na začátku iteračního algoritmu se zvolí stacionární strategie $\bar{\rho}^{(0)}$. Otázkou zůstává výpočet podmíněné střední hodnoty Bellmanovy funkce v rovnici (6.28), resp. (6.32).

Diskrétní hyperstav $\bar{\zeta}'$ odpovídající (6.29) je náhodná proměnná, jelikož hyperstav $\bar{\zeta}'$ je ovlivněn náhodným stavovým šumem. K výpočtu střední hodnoty

$$\mathbb{E} \left\{ \bar{V}^{(i)}(\bar{\zeta}') | \bar{\zeta}, \mathbf{u} \right\} = \mathbb{E} \left\{ \bar{V}^{(i)}(\boldsymbol{\vartheta}(\boldsymbol{\phi}(\bar{\zeta}, \mathbf{u}, \mathbf{x}'))) | \bar{\zeta}, \mathbf{u} \right\} \quad (6.33)$$

je zapotřebí určit podmíněnou hustotu pravděpodobnosti $p(\mathbf{x}' | \bar{\zeta}, \mathbf{u})$, která může být vyjádřena respektováním formulace úlohy jako

$$p(\mathbf{x}' | \bar{\zeta}, \mathbf{u}) = p(\mathbf{x}_{k+1} | \zeta_k, \mathbf{u}_k) = p(\mathbf{x}_{k+1} | \mathbf{Y}_0^k, \mathbf{u}_k). \quad (6.34)$$

Jelikož model (6.29) je nelineární spolu s aproximací Bellmanovy funkce $\bar{V}^{(i)}$, analytický výpočet střední hodnoty není možný. Numerické řešení lze najít použitím některé z metod pravděpodobnosti a statistiky. Pro ukázkou nechť je střední hodnota vypočítána na základě uncentované transformace [104], která využívá pouze první dva momenty náhodné veličiny. Střední hodnota může být aproximována jako

$$\mathbb{E} \left\{ \bar{V}^{(i)}(\boldsymbol{\vartheta}(\boldsymbol{\phi}(\bar{\zeta}, \mathbf{u}, \mathbf{x}'))) | \bar{\zeta}, \mathbf{u} \right\} = \sum_{r=1}^{N_\mu} \sum_{q=0}^{2n_x} = \nu_q \bar{m}_{r,q} P(\mu_k = r | \mathbf{Z}_0^k), \quad (6.35)$$

kde ν_q jsou váhy a $\bar{m}_{r,q}$ jsou transformované σ -body definované na základě σ -bodů $\mathbf{m}_{r,q}$ jako

$$\bar{m}_{r,q} = \bar{V}^{(i)}(\boldsymbol{\vartheta}(\boldsymbol{\phi}(\bar{\boldsymbol{\zeta}}, \mathbf{u}, \mathbf{m}_{r,q}))). \quad (6.36)$$

Váhy ν_q a σ -body $\mathbf{m}_{r,q}$ jsou obecně definovány jako

$$\mathbf{m}_{r,0} = \mathbf{f}_r(\mathbf{x}, \mathbf{u}) + \mathbf{m}^w(\mathbf{x}, r), \quad \nu_0 = \frac{\kappa}{n_x + \kappa}, \quad (6.37)$$

$$\mathbf{m}_{r,\bar{q}} = \mathbf{m}_{r,0} - \left(\sqrt{(n_x + \kappa) \mathbf{G}_r \mathbf{G}_r^T} \right)_{\bar{q}}, \quad \nu_{\bar{q}} = \frac{1}{2(n_x + \kappa)}, \quad (6.38)$$

$$\mathbf{m}_{r,\bar{\bar{q}}} = \mathbf{m}_{r,0} + \left(\sqrt{(n_x + \kappa) \mathbf{G}_r \mathbf{G}_r^T} \right)_{\bar{\bar{q}}-n_x}, \quad \nu_{\bar{\bar{q}}} = \frac{1}{2(n_x + \kappa)}, \quad (6.39)$$

kde $r = 1, 2, \dots, N_\mu$, $\bar{q} = 1, 2, \dots, n_x$, $\bar{\bar{q}} = n_x + 1, n_x + 2, \dots, 2n_x$, \mathbf{x} obsahuje prvních n_x prvků odpovídajícího diskrétního hyperstavu $\bar{\boldsymbol{\zeta}}$, $\kappa \in \mathbb{R}$ je návrhový parametr unscenované transformace a výraz $(\sqrt{(n_x + \kappa) \mathbf{G}_r \mathbf{G}_r^T})_{\bar{q}}$ představuje \bar{q} -tý sloupec matice $\sqrt{(n_x + \kappa) \mathbf{G}_r \mathbf{G}_r^T}$.

Postup návrhu aktivního detektoru chyb pomocí iterace uživatelské funkce je analogický postupu pro iteraci strategie, z toho důvodu je v práci prezentován jen první zmíněný. Celý postup iterace uživatelské funkce s přímou kvantizací spojitého stavového prostoru pro úlohu aktivní detekce chyb je shrnut v algoritmu 5. Důležitým faktorem přesnosti výsledného řešení je vhodný výběr diskrétní množiny hyperstavů \mathcal{R}^g . Iterační algoritmus může být ukončen, pokud je splněna následující podmínka na přesnost aproximace

$$\|\bar{V}^{(i+1)} - \bar{V}^{(i)}\|_\infty \leq \epsilon_{VI} \quad (6.40)$$

kde $\epsilon_{VI} > 0$ je zvolená mez přesnosti.

Algoritmus 5 Suboptimální algoritmus iterace uživatelské funkce s přímou kvantizací spojitého stavového prostoru pro úlohu aktivní detekce chyb.

- 1. Kvantizace prostoru hyperstavů** Volba množiny \mathcal{R}^g diskrétních hyperstavů (6.1).
 - 2. Počáteční podmínky** Inicializace Bellmanovy funkce $\bar{V}^{(0)}$, volba meze přesnosti ϵ_{VI} a nastavení iteračního indexu $i = 0$.
 - 3. Ohodnocení strategie** Výpočet nových hodnot aproximace Bellmanovy funkce $\bar{V}^{(i+1)}(\bar{\boldsymbol{\zeta}})$ pro všechna $\bar{\boldsymbol{\zeta}} \in \mathcal{R}^g$ podle (6.28) s využitím agregační funkce (6.2) a aproximace podmíněné střední hodnoty (6.35).
 - 4. Ukončovací podmínka** Jestliže není splněna ukončovací podmínka (6.40), nastavení $i \leftarrow i + 1$ a **skok zpět na krok 3**.
-

V podkapitolách 6.2 a 6.3 bylo představeno aproximativní řešení úlohy aktivní detekce chyb, které je založené na reprezentaci Bellmanovy funkce formou tabulky. Řešení bylo prezentováno v kontextu úlohy s přímo měřitelnou spojitou částí stavu z důvodu menší dimenze hyperstavu, která je přijatelnější pro kvantizační techniky. Nicméně je možné uvedené řešení zobecnit také pro úlohu s nepřímo měřitelnou spojitou částí stavu. Následující části

se zabývají hledáním řešení pro případ lineární aproximace Bellmanovy funkce. Tato reprezentace Bellmanovy funkce je vhodnější pro stav s velkou dimenzí, a proto je v dalších podkapitolách uvažována úloha s nepřímo měřitelnou spojitou částí stavu, kdy informace o stavu systému je obsažena v hyperstavu $\xi \in \mathcal{G}$.

6.4 Učení temporální diferencí pomocí lineární parametrické aproximace

K řešení komplexních optimalizačních úloh lze kromě aproximačních metod dynamického programování využít zpětnovazební učení [93], tedy typ strojového učení, který vznikl v oblastech informatiky a řízení. Na rozdíl od dynamického programování, které využívá znalost modelu systému, umožňuje zpětnovazební učení hledání řešení optimalizačních úloh na základě vstupně-výstupních dat, které jsou získány z reálného systému nebo simulace jeho chování. Klíčovou vlastností je tedy přirozená identifikace stavového prostoru, kde se stav systému po většinu doby pohybuje, což vede k nalezení přesnější aproximace Bellmanovy funkce na úkor kvality aproximace v částech prostoru, který je méně podstatný. Jelikož provozování systému v chybovém stavu je nežádoucí, je v případě detekce chyb vhodnější použít pro zpětnovazební učení data simulovaná pomocí modelu systému.

Často používané metody zpětnovazebního učení sloužící k nalezení regulátoru v úloze optimálního řízení systému jsou založeny na technice *učení temporální diferencí* (TD) [105], [106]. Učení TD zaznamenalo výrazný úspěch při návrhu počítačového programu, který byl schopen porazit nejlepší světové hráče staré deskové hry vrhcáby, známé pod anglickým názvem backgammon [107], [108]. Optimální Bellmanova funkce ve hře vrhcáby byla aproximována neuronovou sítí, která byla trénována simulacemi hry a učením TD. Jedná se o první úspěšnou aplikaci této techniky na komplexní úlohu. Mezi další deskové hry, kde bylo učení TD zdárně aplikováno, patří dáma [109] nebo šachy [110]. Následující návrh aktivního detektoru chyb využívá učení TD a často používanou lineární parametrickou aproximaci Bellmanovy funkce, která bude nejprve přiblížena.

6.4.1 Lineární parametrická aproximace

Parametrické funkční aproximace představují zobrazení z prostoru parametrů do prostoru funkcí aproximujících hledanou optimální Bellmanovu funkci. Počet parametrů spolu s architekturou funkcí je předem známý. Z důvodu obecnosti je v této části vynechán časový index. Necht $\tilde{V} : \mathcal{G} \times \mathbb{R}^{n_\omega} \mapsto \mathbb{R}$ je parametrická aproximace Bellmanovy funkce, která závisí na hyperstavu $\xi \in \mathcal{G}$ a na vektoru parametrů $\omega \in \mathbb{R}^{n_\omega}$, který je také nazýván jako vektor vah. Při použití parametrické aproximace \tilde{V} není zapotřebí znát hodnotu Bellmanovy funkce pro všechny možné hyperstavy $\xi \in \mathcal{G}$, ale stačí pouze uchovávat hodnoty n_ω parametrů. Právě volba rozumného počtu parametrů poskytuje kompaktní reprezentaci hledané funkce.

Parametrické funkční aproximace mohou být lineární nebo nelineární v parametrech. Významnou skupinu tvoří lineární aproximace, jelikož jejich struktura umožňuje lepší ana-

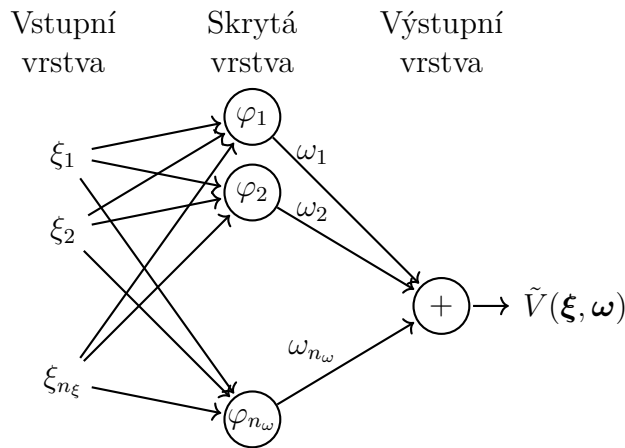
lýzu teoretických vlastností a konvergence. Za určitých podmínek lze Bellmanovu funkci bez ztráty obecnosti nahradit jako

$$V(\boldsymbol{\xi}) = \sum_{j=1}^{\infty} \varphi_j(\boldsymbol{\xi})\omega_j = \sum_{j=1}^{n_\omega} \varphi_j(\boldsymbol{\xi})\omega_j + \sum_{j=n_\omega+1}^{\infty} \varphi_j(\boldsymbol{\xi})\omega_j = \boldsymbol{\varphi}^T(\boldsymbol{\xi})\boldsymbol{\omega} + \epsilon_{n_\omega}(\boldsymbol{\xi}), \quad (6.41)$$

kde $\boldsymbol{\varphi} : \mathcal{G} \mapsto \mathbb{R}^{n_\omega}$ je vektorová funkce vektorového argumentu, $\boldsymbol{\omega} = [\omega_1, \dots, \omega_{n_\omega}]^T$ je vektor vah a $\epsilon_{n_\omega} : \mathcal{G} \mapsto \mathbb{R}^{n_\omega}$ je člen stejnoměrně konvergující do nuly pro $n_\omega \rightarrow \infty$ [79]. Člen $\boldsymbol{\varphi}$ bývá označován jako vektor bázových funkcí, jelikož jeho prvky jsou tvořeny pevnými známými bázovými funkcemi $\varphi_j : \mathcal{G} \mapsto \mathbb{R}$, $\boldsymbol{\varphi} = [\varphi_1(\boldsymbol{\xi}), \dots, \varphi_{n_\omega}(\boldsymbol{\xi})]^T$. Z důvodu zachování přiměřených výpočetních a paměťových nároků je zapotřebí vhodně zvolit konečné číslo n_ω . Bellmanova funkce je aproximována lineární parametrickou funkcí $\tilde{V} : \mathcal{G} \times \mathbb{R}^{n_\omega} \mapsto \mathbb{R}$ tak, že

$$\tilde{V}(\boldsymbol{\xi}, \boldsymbol{\omega}) = \sum_{j=1}^{n_\omega} \varphi_j(\boldsymbol{\xi})\omega_j = \boldsymbol{\varphi}^T(\boldsymbol{\xi})\boldsymbol{\omega}, \quad (6.42)$$

kde $\boldsymbol{\omega} \in \mathbb{R}^{n_\omega}$ je neznámý vektor parametrů, jehož hodnoty definují Bellmanovu funkci a nalezení funkce je tak převedeno na nalezení hodnot parametrů. Bázové funkce musí být vhodně zvoleny apriori na základě simulování systému, zkušenosti nebo znalosti úlohy. Jelikož architektura definovaná v (6.42) by měla být schopna dostatečně aproximovat skutečnou Bellmanovu funkci pouze použitím n_ω parametrů, počet bázových funkcí a jejich výběr je jedním z rozhodujících faktorů ovlivňujících kvalitu aproximativního řešení. Lineární parametrická aproximace (6.42) může být chápána jako neuronová síť, jak je ilustrováno na obrázku 6.7. Vstupní vrstvu tvoří jednotlivé složky hyperstavu, skrytá vrstva se skládá z bázových funkcí a výstupem neuronové sítě je aproximace Bellmanovy funkce. Jednotlivé váhy poté řídí zobrazení ze skryté vrstvy do výstupní vrstvy.



Obrázek 6.7: Ilustrace struktury lineární parametrické aproximace (6.42) jako neuronové sítě.

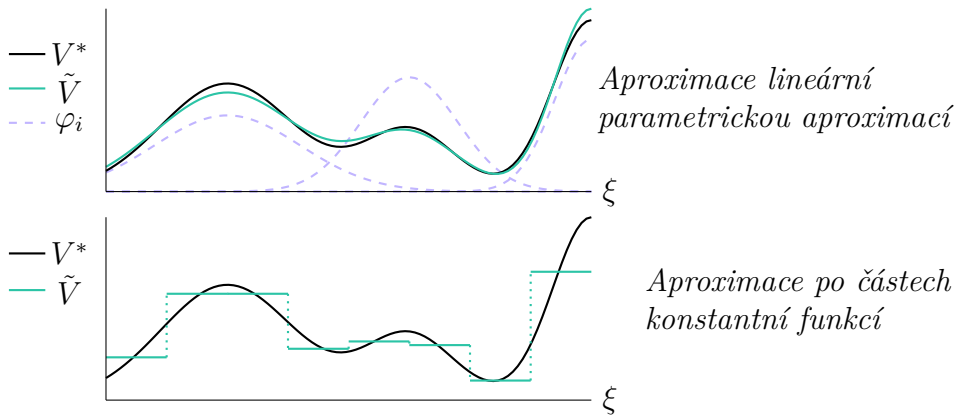
Mezi základní běžně používané bázové funkce patří polynomy, trigonometrické funkce nebo radiální bázové funkce. Radiální bázové funkce jsou funkce reálných hodnot, které závisí pouze na vzdálenosti od jejich středu [111]. Často používaná normalizovaná Gaussova radiální bázová funkce může být v kontextu aktivní detekce chyb definována jako

$$\varphi_j(\boldsymbol{\xi}) = \frac{\exp\left(-\frac{\|\boldsymbol{\xi}-\boldsymbol{\xi}_j^c\|_2^2}{\delta_j^\xi}\right)}{\sum_{i=1}^{n_\omega} \exp\left(-\frac{\|\boldsymbol{\xi}-\boldsymbol{\xi}_i^c\|_2^2}{\delta_i^\xi}\right)}, \quad (6.43)$$

kde $\boldsymbol{\xi}_j^c \in \mathcal{G}$ je střed j -té radiální bázové funkce a $\delta_j^\xi > 0$ je návrhový parametr udávající rozpětí funkce. Další možnou formu bázové funkce představuje po částech konstantní funkce, která se využívá při kvantizaci spojitého prostoru stavů. Při vhodné volbě agregační funkce $\boldsymbol{\vartheta} : \mathcal{G} \mapsto \bar{\mathcal{G}}$ zobrazující hyperstav $\boldsymbol{\xi} \in \mathcal{G}$ do diskrétního hyperstavu $\bar{\boldsymbol{\xi}} \in \bar{\mathcal{G}}$ z diskrétní množiny hyperstavů $\bar{\mathcal{G}} = \{\bar{\boldsymbol{\xi}}^1, \bar{\boldsymbol{\xi}}^2, \dots, \bar{\boldsymbol{\xi}}^{N_\xi}\}$, může být spojitý prostor hyperstavů kvantizován. Odpovídající tvar bázové funkce může být definován jako

$$\varphi_j(\boldsymbol{\xi}) = \begin{cases} 1 & \text{pokud } \boldsymbol{\vartheta}(\boldsymbol{\xi}) = \bar{\boldsymbol{\xi}}^j, \\ 0 & \text{jinak,} \end{cases} \quad (6.44)$$

kde $j = 1, 2, \dots, N_\xi$ a počet vah odpovídá počtu diskrétních stavů $n_\omega = N_\xi$. Nicméně tento postup se stává výpočetně náročný při požadavku na jemnou kvantizaci, jež způsobí velkou dimenzi diskrétního prostoru. Tato forma aproximace byla diskutována v podkapitolách 6.2 a 6.3. Porovnání aproximace Bellmanovy funkce lineární parametrickou aproximací s normalizovanými gaussovskými bázovými funkcemi a aproximace po částech konstantní funkcí je ilustrováno na obrázku 6.8.



Obrázek 6.8: Ilustrace rozdílů mezi aproximací Bellmanovy funkce lineární parametrickou aproximací tvořenou normalizovanými gaussovskými bázovými funkcemi a aproximací po částech konstantní funkcí v jednodimenzionálním stavovém prostoru.

6.4.2 Učení temporální diferencí

Učení TD se podobně jako uvedené iterační algoritmy dynamického programování skládá ze dvou kroků: ohodnocení strategie a vylepšení strategie. V kontextu úlohy aktivní detekce chyb používá učení TD na rozdíl od dynamického programování pro ohodnocení strategie data. Nicméně pro vylepšení strategie, přesněji pro generování vstupu, je zapotřebí znát model systému v důsledku přeformulování na úlohu s úplnou informací o stavu. Tato část pojednává o možném způsobu nalezení vektoru vah ω lineární parametrické aproximace Bellmanovy funkce pomocí techniky učení TD. V každé iteraci je vektor vah aktualizován na základě získaných dat. Iterační index TD $m = 0, 1, 2, \dots$ může jít s časem, neboli v každém časovém okamžiku je provedena právě jedna iterace, a nebo lze realizovat několik iterací vylepšení vektoru vah v rámci jednoho časového okamžiku. Z důvodu obecnosti je v následujících vztazích zachován iterační index m .

Nechť je uvažována m -tá iterace algoritmu, která využívá vzorek hyperstavu ξ_{k+1} generovaného na základě aktuálně platné strategie $\tilde{\rho}^{(m)}$ a vzorku hyperstavu ξ_k . To znamená, že rozhodnutí d_k a vstup \mathbf{u}_k jsou generovány podle

$$\begin{bmatrix} d_k \\ \mathbf{u}_k \end{bmatrix} = \tilde{\rho}^{(m)}(\xi_k). \quad (6.45)$$

Strategie $\tilde{\rho}^{(m)} = [\bar{\sigma}^*, (\tilde{\gamma}^{(m)})^T]^T$ aproximující Bellmanovu funkci je stanovena na základě vah $\omega^{(m)}$ jako

$$\bar{\sigma}^*(\xi_k) = \arg \min_{d \in \mathcal{M}} \bar{L}^d(\xi_k, d), \quad (6.46)$$

$$\tilde{\gamma}^{(m)}(\xi_k) = \arg \min_{\mathbf{u} \in \mathcal{U}} E \left\{ \tilde{V}(\phi(\xi_k, \mathbf{u}, \mathbf{y}'), \omega^{(m)}) \mid \xi_k, \mathbf{u} \right\}, \quad (6.47)$$

kde generátor rozhodnutí $\bar{\sigma}^*$ je vypočítán přímo na základě znalosti funkce \bar{L}^d a generátor vstupního signálu $\tilde{\gamma}^{(m)}$ je v každé iteraci m aktualizován na základě dostupných dat. Rovnice (6.46) a (6.47) představují krok vylepšení strategie s využitím aproximace Bellmanovy funkce (6.42). Generátor rozhodnutí se nemění na rozdíl od generátoru vstupního signálu, který závisí na dané lineární parametrické aproximaci Bellmanovy funkce. Je nutné zdůraznit, že vztah (6.47) vyžaduje explicitní znalost modelu ϕ a měřený výstup \mathbf{y}' je použitý k výpočtu hyperstavu ξ_{k+1} pomocí (5.54). Funkce $\tilde{V}^{\tilde{\rho}^{(m-1)}}$ definovaná jako

$$\tilde{V}^{\tilde{\rho}^{(m-1)}}(\xi) = \tilde{V}(\xi, \omega^{(m)}) \quad (6.48)$$

představuje Bellmanovu funkci pro suboptimální strategii $\tilde{\rho}^{(m-1)}$ z předchozího iteračního kroku. V případě počáteční fáze iteračního algoritmu pro $m = 0$ může být počáteční vektor vah $\omega^{(0)}$ zvolen libovolně [106]. V případě simulace systému na základě modelu chování systému (5.54) může být podmíněná střední hodnota v (6.47) vyhodnocena přibližně pomocí metody MC.

Krok ohodnocení strategie spočívá v případě dynamického programování v nalezení funkce $\tilde{V}^{\tilde{\rho}^{(m)}}$ pro danou strategii $\tilde{\rho}^{(m)}$, neboli nalezení hodnot pro všechny možné hyperstavy $\xi \in \mathcal{G}$. V případě učení TD není krok ohodnocení strategie proveden úplně,

jelikož jsou dostupné pouze dva vzorky hyperstavu. V m -té iteraci je nový vektor vah $\omega^{(m+1)}$ získán na základě znalosti vektoru vah $\omega^{(m)}$ a vzorků hyperstavu ξ_k a ξ_{k+1} . K tomu slouží tzv. chyba temporální difference (CHTD), která reprezentuje rozdíl mezi očekávanými ztrátami $\tilde{V}(\xi_k, \omega^{(m)})$ a predikovanými ztrátami $\bar{L}^d(\xi_k, d_k) + \eta \tilde{V}(\xi_{k+1}, \omega^{(m)})$ na základě simulovaných dat. Hodnota CHTD δ_k je proto definována jako

$$\delta_k = \bar{L}^d(\xi_k, d_k) + \eta \tilde{V}(\xi_{k+1}, \omega^{(m)}) - \tilde{V}(\xi_k, \omega^{(m)}). \quad (6.49)$$

V kroku ohodnocení strategie jsou tedy váhy aktualizovány o informaci z CHTD a nový vektor vah je vypočítán jako

$$\omega^{(m+1)} = \omega^{(m)} + \kappa_k^t \delta_k \sum_{t=0}^k (\eta \lambda)^{k-t} \varphi(\xi_t), \quad (6.50)$$

kde $\lambda \in \langle 0, 1 \rangle$ je parametr TD a $\kappa_k^t > 0$ je parametr délky kroku, který se může v čase měnit. Na základě parametru TD λ jsou takové metody nazývané jako TD(λ) metody učení. Vztah (6.50) může být zapsán v kompaktnější podobě jako

$$\omega^{(m+1)} = \omega^{(m)} + \kappa_k^t \delta_k \mathbf{z}_k, \quad (6.51)$$

kde $\mathbf{z}_k \in \mathbb{R}^{n_\omega}$ je tzv. vektor zodpovědnosti, který je rekurzivně vyjádřen jako

$$\mathbf{z}_{k+1} = \eta \lambda \mathbf{z}_k + \varphi(\xi_{k+1}) \quad (6.52)$$

s počáteční podmínkou $\mathbf{z}_0 = \varphi(\xi_0)$. Vektor zodpovědnosti \mathbf{z}_k poskytuje informaci o tom, které z vah jsou zodpovědné za současnou CHTD, a proto musí být pozměněny.

Jelikož aktualizace vah probíhá na základě na dat, je zapotřebí dostatečně prohledávat prostor hyperstavů. V případě nedostatečného prohledání důležitých regionů hyperstavů může výsledná strategie generovat vstup, který nebude pomáhat ke zlepšení detekce chyb. Jednou z možností je změna výběru počátečního hyperstavu tak, aby došlo k co nejlepšímu pokrytí prostoru hyperstavů. Další techniky jsou diskutovány v podkapitole 6.4.3, kde je v průběhu učení zapotřebí prohledávat jak prostor hyperstavů, tak prostor vstupů.

Postup učení TD pro úlohu aktivní detekce chyb je shrnut v algoritmu 6. Uvedený algoritmus představuje variantu, kdy je v každém časovém okamžiku provedena právě jedna iterace ohodnocení strategie, neboli index m jde s časovým indexem k . V případě dostupnosti dat ze dvou časových okamžiků je možné aktuální strategii ohodnotit a následně vylepšit na základě nalezeného vektoru vah. Kroky iteračního algoritmu učení TD jsou opakovány, dokud není splněna zvolená ukončovací podmínka. Taková podmínka může spočívat v dosažení maximálního počtu iteračních kroků, nebo například ve splnění určitého kritéria kvality pro aproximaci Bellmanovy funkce. Nicméně je nutné zdůraznit, že v každém iteračním kroku vylepšení strategie (6.47) je zapotřebí vypočítat jednu střední hodnotu očekávaných ztrát pro všechny možné vstupy z množiny všech přípustných vstupů \mathcal{U} . Navíc tato střední hodnota nemůže být vyjádřena analyticky díky nelinearitě v modelu systému, a proto je výpočet nutné provést aproximačně, například uncentrovanou transformací nebo metodou MC. Takový výpočet však může být velmi výpočetně náročný. Alternativní způsob využívající tzv. Q-funkci těmito výpočetně náročným postupům předejde na úkor zvýšení paměťových nároků.

Algoritmus 6 Algoritmus učení TD pro úlohu aktivní detekce chyb.

Inicializace Výběr počátečního vektoru vah $\omega^{(0)}$, parametrů κ_k^t , λ a počtu MC simulací N_{MC} pro výpočet střední hodnoty v (6.47). Nastavení iteračního indexu $m = 0$ a časového indexu $k = 0$.

1. Pozorování Měření výstupu \mathbf{y}_k a sestavení hyperstavu ξ_k pomocí algoritmu odhadu stavu a přeformulování úlohy.

2. Ohodnocení strategie Pokud $k \geq 1$, následuje výpočet vektoru vah $\omega^{(m)}$ využitím učení TD.

2.1) Sestavení ξ_{k-1} a ξ_k .

2.2) Výpočet CHTD δ_{k-1} pomocí (6.49).

2.3) Výpočet vektoru vah $\omega^{(m+1)}$ na základě (6.51) a (6.52).

2.4) Nastavení $m \leftarrow m + 1$.

3. Rozhodnutí a vstup Generování rozhodnutí d_k a vstupu \mathbf{u}_k na základě průběžné strategie $\tilde{\rho}^{(m)}$ (6.45) s využitím (5.54), (6.42), (6.46), (6.47) a aktuálního vektoru vah $\omega^{(m)}$.

Nastavení $k \leftarrow k + 1$ a skok zpět na krok 1 dokud není splněna ukončovací podmínka.

6.4.3 Q-učení temporální diferencí

Q-učení temporální diferencí (QTD) je technika velmi podobná učení TD, avšak místo Bellmanovy funkce je uvažována Q-funkce. Q-funkce je funkce, která se velmi často využívá ve zpětnovazebním učení i dynamickém programování. Jejím použitím lze generovat vstupní signál bez znalosti modelu systému. Na rozdíl od Bellmanovy funkce je kromě stavu Q-funkce závislá také na vstupu. V problematice aktivní detekce chyb lze optimální *Q-funkci* [112] definovat jako očekávané diskontované ztráty, které by nastaly v případě zvolení rozhodnutí $d \in \mathcal{M}$ a vstupu $\mathbf{u} \in \mathcal{U}$ pro hyperstav $\xi \in \mathcal{G}$ a v následujících časových okamžicích by byla používána optimální strategie $\bar{\rho}^*$, neboli

$$Q^*(\xi, \mathbf{u}, d) = \bar{L}^d(\xi, d) + \eta \mathbb{E} \{V^*(\phi(\xi, \mathbf{u}, \mathbf{y}')) | \xi, \mathbf{u}\}, \quad (6.53)$$

kde $Q^* : \mathcal{G} \times \mathcal{U} \times \mathcal{M} \mapsto \mathbb{R}$ je optimální Q-funkce hyperstavu ξ , vstupu \mathbf{u} a rozhodnutí d . Z důvodu zjednodušení byly v (6.53) vynechány časové indexy. Optimální aktivní detektor chyb $\bar{\rho}^* = [\bar{\sigma}^*, (\bar{\gamma}^*)^T]^T$ lze v každém časovém okamžiku $k \in \mathcal{T}$ zapsat pomocí Q-funkce jako

$$\bar{\sigma}^*(\xi_k) = \arg \min_{d \in \mathcal{M}} \bar{L}^d(\xi_k, d), \quad (6.54)$$

$$\bar{\gamma}^*(\xi_k) = \arg \min_{\mathbf{u} \in \mathcal{U}} \{Q^*(\xi_k, \mathbf{u}, \bar{\sigma}^*(\xi_k))\}. \quad (6.55)$$

Generátor vstupního signálu je opět plně definován ztrátovou funkcí \bar{L}^d , avšak ke generování vstupu je zapotřebí najít optimální Q-funkci Q^* . Stojí za povšimnutí, že vztah (6.55) na rozdíl od vztahu (6.47) neobsahuje výpočet střední hodnoty a nevyžaduje tak znalost modelu. Libovolná strategie $\bar{\rho} = [\bar{\sigma}, (\bar{\gamma})^T]^T$ může být ohodnocena s využitím Q-funkce následovně

$$Q^{\bar{\rho}}(\boldsymbol{\xi}, \mathbf{u}, d) = \bar{L}^d(\boldsymbol{\xi}, d) + \eta \mathbb{E}\{Q^{\bar{\rho}}(\phi(\boldsymbol{\xi}, \mathbf{u}, \mathbf{y}'), \bar{\gamma}(\phi(\boldsymbol{\xi}, \mathbf{u}, \mathbf{y}'))), \bar{\sigma}(\phi(\boldsymbol{\xi}, \mathbf{u}, \mathbf{y}')) | \boldsymbol{\xi}, \mathbf{u}\}. \quad (6.56)$$

Z této rovnice je patrné, že Q-funkce odpovídá součtu bezprostředních ztrát $\bar{L}^d(\boldsymbol{\xi}, d)$ závislejších na rozhodnutí d a nějaké funkci budoucích ztrát, která závisí na hyperstavu $\boldsymbol{\xi}$ a vstupu \mathbf{u} . Q-funkci není jednoduché vypočítat v důsledku prokletí rozměrnosti. Tento jev, který nastává především v důsledku vlastností hyperstavu $\boldsymbol{\xi}$ je důvodem pro nahrazení Q-funkce nějakou vhodnou aproximací. V případě volby lineární parametrické aproximace je možné Q-funkci aproximovat jako

$$\begin{aligned} \tilde{Q}(\boldsymbol{\xi}, \mathbf{u}, d, \boldsymbol{\theta}) &= \bar{L}^d(\boldsymbol{\xi}, d) + \sum_{i=1}^{n_\theta} \chi_i(\boldsymbol{\xi}, \mathbf{u}) \theta_i \\ &= \bar{L}^d(\boldsymbol{\xi}, d) + \boldsymbol{\chi}(\boldsymbol{\xi}, \mathbf{u})^T \boldsymbol{\theta}, \end{aligned} \quad (6.57)$$

kde $\tilde{Q} : \mathcal{G} \times \mathcal{U} \times \mathcal{M} \times \mathbb{R}^{n_\theta} \mapsto \mathbb{R}$ je aproximace Q-funkce, $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ je vektor vah Q-funkce a $\boldsymbol{\chi} : \mathcal{G} \times \mathcal{U} \mapsto \mathbb{R}^{n_\theta}$ je vektor bázových funkcí. Vektor vah je opět tvořen jednotlivými neznámými parametry $\boldsymbol{\theta} = [\theta_1, \dots, \theta_{n_\theta}]^T$ a vektor bázových funkcí $\boldsymbol{\chi}(\boldsymbol{\xi}, \mathbf{u}) = [\chi_1(\boldsymbol{\xi}, \mathbf{u}), \dots, \chi_{n_\theta}(\boldsymbol{\xi}, \mathbf{u})]^T$ známými funkcemi, které jsou vhodně zvoleny na základě zkušenosti nebo porozumění dané úloze. Pohledem na vztah (6.57) lze zjistit dva hlavní rozdíly od lineární parametrické aproximace pro Bellmanovu funkci uvedené v (6.42). V případě Q-funkce jsou bázové funkce kromě hyperstavu $\boldsymbol{\xi}$ závislé také na vstupu \mathbf{u} . Druhým rozdílem je explicitní vyjádření členu $\bar{L}^d(\boldsymbol{\xi}, d)$, které plyne z definice Q-funkce. Jelikož hodnota ztrát $\bar{L}^d(\boldsymbol{\xi}, d)$ je známá, lze lineární parametrickou funkcí aproximovat pouze druhý člen ve vztahu (6.56). Úkolem zpětnovazebního učení je najít vektor vah $\boldsymbol{\theta}$ tak, aby funkce \tilde{Q} co nejlépe aproximovala optimální Q-funkci Q^* .

Následující část přibližuje učení QTD. Níže uvedené vztahy jsou proto velmi podobné těm, které byly prezentovány v podkapitole 6.4.2. Iterační krok vylepšení strategie $\tilde{\rho}^{(m)} = [\bar{\sigma}^*, (\tilde{\gamma}^{(m)})^T]^T$ pomocí QTD odpovídá

$$\bar{\sigma}^*(\boldsymbol{\xi}_k) = \arg \min_{d \in \mathcal{M}} \bar{L}^d(\boldsymbol{\xi}_k, d), \quad (6.58)$$

$$\tilde{\gamma}^{(m)}(\boldsymbol{\xi}_k) = \arg \min_{\mathbf{u} \in \mathcal{U}} \left\{ \tilde{Q}(\boldsymbol{\xi}_k, \mathbf{u}, \bar{\sigma}^*(\boldsymbol{\xi}_k), \boldsymbol{\theta}^{(m)}) \right\}, \quad (6.59)$$

kde aproximace Q-funkce $\tilde{Q}^{\tilde{\rho}^{(m-1)}}$ pro strategii $\tilde{\rho}^{(m-1)}$ je definována jako

$$\tilde{Q}^{\tilde{\rho}^{(m-1)}}(\boldsymbol{\xi}, \mathbf{u}, \bar{\sigma}^*(\boldsymbol{\xi})) = \tilde{Q}(\boldsymbol{\xi}, \mathbf{u}, \bar{\sigma}^*(\boldsymbol{\xi}), \boldsymbol{\theta}^{(m)}). \quad (6.60)$$

Uvedené vztahy podobně jako v učení TD obsahují obecný iterační index $m = 0, 1, 2, \dots$, který může jít s časem; další možností je realizace několika iterací vylepšení vektoru vah v jednom časovém okamžiku. Druhým krokem učení QTD je ohodnocení strategie $\tilde{\rho}^{(m)}$ s využitím Q-funkce $\tilde{Q}^{\tilde{\rho}^{(m)}}$, vektoru vah $\boldsymbol{\theta}^{(m)}$ a CHTD pro Q-funkci δ_k^Q definované jako

$$\begin{aligned} \delta_k^Q &= \bar{L}^d(\boldsymbol{\xi}_k, d_k) + \eta \tilde{Q}(\boldsymbol{\xi}_{k+1}, \mathbf{u}_{k+1}, \bar{\sigma}^*(\boldsymbol{\xi}_{k+1}), \boldsymbol{\theta}^{(m)}) \\ &\quad - \tilde{Q}(\boldsymbol{\xi}_k, \mathbf{u}_k, \bar{\sigma}^*(\boldsymbol{\xi}_k), \boldsymbol{\theta}^{(m)}). \end{aligned} \quad (6.61)$$

Nový vektor vah Q-funkce je tedy získán, obdobně jako v (6.50), podle

$$\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} + \kappa_k^q \delta_k^Q \sum_{t=0}^k (\eta\lambda)^{k-t} \boldsymbol{\chi}(\boldsymbol{\xi}_t, \mathbf{u}_t), \quad (6.62)$$

kde $\kappa_k^q > 0$ je parametr délky kroku. Zavedením vektoru zodpovědnosti $\mathbf{z}_k^Q \in \mathbb{R}^{n_\theta}$ definovaného jako

$$\mathbf{z}_{k+1}^Q = \eta\lambda \mathbf{z}_k^Q + \boldsymbol{\chi}(\boldsymbol{\xi}_{k+1}, \mathbf{u}_{k+1}) \quad (6.63)$$

může být vztah (6.62) přepsán do rekurzivní podoby

$$\boldsymbol{\theta}^{(m+1)} = \boldsymbol{\theta}^{(m)} + \kappa_k^q \delta_k^Q \mathbf{z}_k^Q \quad (6.64)$$

s volbou počáteční podmínky $\mathbf{z}_0^Q = \boldsymbol{\chi}(\boldsymbol{\xi}_0, \mathbf{u}_0)$.

V případě učení QTD je zapotřebí dostatečně prohledat jak stavový prostor, tak prostor možných budících vstupů. Z toho důvodu by se měla do procesu generování vstupu zavést náhoda. Jednou z možností dosažení potřebných vlastností je zavedení tzv. ϵ -hladové strategie. S pravděpodobností $(1 - \epsilon)$ se použije vstup generovaný na základě (6.59) a s pravděpodobností ϵ je vstup vygenerován náhodně z množiny všech přípustných vstupů při použití rovnoměrného rozdělení. Nedostatkem ϵ -hladové strategie je to, že při generování náhodného vstupu jsou všechny možné vstupy vážené stejně. Jako lepší postup by se zdál výběr patrně lepších vstupů než těch, které v dané situaci působí méně slibně. Takový výběr může být dosažen metodami, které jsou známy jako soft-min, případně soft-max metody [93]. Základní postup nalezení aktivního detektoru chyb pomocí QTD a ϵ -hladové strategie je uveden v algoritmu 7.

Oba algoritmy 6 a 7 představují možný způsob hledání vhodné strategie aktivního detektoru chyb založené na off-line simulaci modelu systému. To znamená, že proces vylepšování strategie probíhá nad vzorky ze simulátoru systému, jelikož spuštění skutečného systému by mohlo vést k nežádoucímu chování v důsledku chyby v systému. Off-line navržený aktivní detektor chyb splňující určité požadavky na chování a podmínky provozu je následně využit on-line v reálném systému, jelikož výpočetní čas generování rozhodnutí a budícího vstupu je malý, zvláště pak v případě použití Q-funkce.

Algoritmus 7 Algoritmus QTD využívající ϵ -hladovou strategii pro úlohu aktivní detekce chyb.

Inicializace Výběr vektoru vah $\boldsymbol{\theta}^{(0)}$, básových funkcí $\boldsymbol{\chi}$, parametrů κ_k^q , λ a ϵ . Nastavení iteračního indexu $m = 0$ a časového indexu $k = 0$.

1. Pozorování Měření výstupu \mathbf{y}_k a sestavení hyperstavu $\boldsymbol{\xi}_k$ pomocí algoritmu odhadu stavu a přeformulování úlohy.

2. Hledání strategie Pokud $k \geq 1$, následuje výpočet vektoru vah $\boldsymbol{\theta}^{(m)}$ využitím QTD.

2.1) Načtení $\boldsymbol{\xi}_{k-1}$ a $\boldsymbol{\xi}_k$.

2.2) Výpočet CHTD δ_{k-1}^Q pomocí (6.61).

2.3) Výpočet vektoru vah $\boldsymbol{\theta}^{(m+1)}$ na základě (6.63) a (6.64).

2.4) Nastavení $m \leftarrow m + 1$.

3. Rozhodnutí a vstup Generování rozhodnutí d_k podle (6.58) a (5.57). S pravděpodobností $1 - \epsilon$ generování vstupu \mathbf{u}_k na základě (6.59), (6.57), (5.54) a současného vektoru vah $\boldsymbol{\theta}^{(m)}$. S pravděpodobností ϵ/N_u náhodný výběr vstupu jako $\mathbf{u}_k = \bar{\mathbf{u}}^i$, $\bar{\mathbf{u}}^i \in \mathcal{U}$ pro $i = 1, \dots, N_u$.

Nastavení $k \leftarrow k + 1$ a skok zpět na krok 1 dokud není splněna ukončovací podmínka.

6.5 Použití neparametrické aproximace

Jedna z nevýhod parametrické aproximace Bellmanovy funkce spočívá v nutnosti dopředné volby struktury aproximace bez možnosti využití naměřených dat. Nevhodná volba struktury zásadně ovlivní výsledek hledání optimálního řešení optimalizační úlohy. Oproti tomu neparametrická aproximace disponuje jistou flexibilitou. Neparametrické aproximace obsahují členy podobné parametrům, jejichž počet a zároveň hodnoty se mohou měnit a závisí na dostupných měřeních. Hlavní myšlenkou neparametrických aproximací je tedy sestavení lokálních aproximací na základě naměřených dat.

V neparametrických architektuurách se velmi často využívá tzv. jádrová funkce [111]. Bellmanova funkce může být aproximována váženým součtem

$$\tilde{V}(\boldsymbol{\xi}) = \frac{\sum_{j=1}^{N_{\xi}^s} K(\boldsymbol{\xi}, \boldsymbol{\xi}_j^s) V_j^s}{\sum_{j=1}^{N_{\xi}^s} K(\boldsymbol{\xi}, \boldsymbol{\xi}_j^s)}, \quad (6.65)$$

kde $\boldsymbol{\xi}_j^s \in \mathcal{G}^s \subset \mathcal{G}$ je vzorek hyperstavu z diskrétní množiny N_{ξ}^s vzorků hyperstavů, V_j^s je přibližná hodnota Bellmanovy funkce pro hyperstav $\boldsymbol{\xi}_j^s$ a $K : \mathcal{G} \times \mathcal{G}^s \mapsto \mathbb{R}$ je jádrová funkce udávající váhu mezi $\boldsymbol{\xi} \in \mathcal{G}$ a $\boldsymbol{\xi}_j^s$, která musí splňovat určité vlastnosti [113]. Hodnoty V_j^s pro všechna $j = 1, 2, \dots, N_{\xi}^s$ mohou být stanoveny off-line nebo on-line na základě měření.

Velmi oblíbenou volbou jádra K je Gaussova jádrová funkce definovaná následovně

$$K(\boldsymbol{\xi}, \boldsymbol{\xi}_i) = \exp\left(-\frac{\|\boldsymbol{\xi} - \boldsymbol{\xi}_i\|_2^2}{\delta^K}\right), \quad (6.66)$$

kde $\delta^K > 0$ je návrhový parametr udávající rozpětí jádrové funkce. Jeho volba ovlivní přesnost výsledného řešení. Přehled dalších používaných jádrových funkcí je uveden například v [114]. Mezi další neparametrické architektury se řadí aproximace využívající gaussovské processy [115]–[117] nebo hierarchické agregace [118].

Flexibilita neparametrických aproximací ovšem ztěžuje analýzu konvergence a v případě velkého počtu naměřených dat může způsobit vysoké výpočetní a paměťové nároky. Složitost aproximace souvisí s její strukturou a počtem parametrů. Z těchto důvodů je zřejmé, že volba struktury a počtu parametrů je stěžejním bodem řešením optimalizační úlohy. Parametrické a neparametrické architektury jsou blíže diskutovány například v [111], [114]. V této části je Bellmanova funkce nahrazena lokální neparametrickou aproximací.

Nechť zvolená aproximace využívá strukturu aproximace definovanou v (6.65) a gaussovskou jádrovou funkci ze vztahu (6.66). K použití zvolené aproximace je zapotřebí definovat diskrétní množinu vzorků hyperstavů \mathcal{G}^s a vypočítat příslušné hodnoty V_j^s pro všechna $j = 1, 2, \dots, N_\xi^s$. Tyto vzorky umožňují jistou flexibilitu aproximace a samotná diskrétní množina $\mathcal{G}^s = \{\boldsymbol{\xi}_1^s, \boldsymbol{\xi}_2^s, \dots, \boldsymbol{\xi}_{N_\xi^s}^s\}$ hyperstavů může být určena na základě zkušeností nebo apriorní znalosti úlohy. Taková definice množiny \mathcal{G}^s však není vhodná pro systémy s vysokou dimenzí hyperstavu. Existují tedy další možnosti zadefinování množiny \mathcal{G}^s . Jednou z možností je vytvoření diskrétní mřížky pokrývající důležité regiony prostoru hyperstavů podobně jako v podkapitolách 6.2 a 6.3. Nevýhodou tohoto postupu jsou vyšší paměťové a výpočetní nároky. Navíc se hyperstav v závislosti na dynamice systému může pohybovat převážně v určitých oblastech a rovnoměrné pokrytí celého prostoru hyperstavů není důležité tak, jako vyšší hustota pokrytí těchto oblastí. Aby se předešlo nedostatkům předchozích postupů, mohou být vzorky hyperstavu tvořící množinu \mathcal{G}^s vygenerovány na základě různorodých simulací systému [119]. To znamená, že vzorky hyperstavů mohou být generovány pomocí vhodně zvolených budicích signálů a získané trajektorie systému. V případě nestabilního systému je nutné zvolit stabilizující vstupní signál.

Za předpokladu známé množiny \mathcal{G}^s následuje druhý krok v podobě výpočtu hodnot Bellmanovy funkce V_j^s pro jednotlivé vzorky hyperstavů $\boldsymbol{\xi}_j^s \in \mathcal{G}^s$. Takový výpočet lze provést pomocí algoritmů dynamického programování, například iteraci užitkové funkce. V každé iteraci $i = 0, 1, 2, \dots$ jsou přibližné hodnoty Bellmanovy funkce nad vzorky hyperstavů vylepšeny následovně

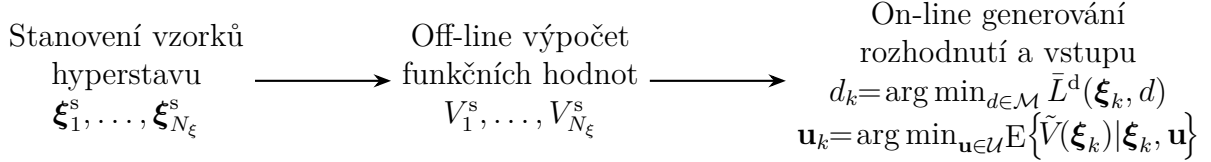
$$V_j^{s(i+1)} = \min_{d \in \mathcal{M}} E \left\{ \bar{L}^d(\boldsymbol{\xi}_j^s, d) | \boldsymbol{\xi}_j^s, d \right\} + \eta \min_{\mathbf{u} \in \mathcal{U}} E \left\{ \tilde{V}^{s(i)}(\boldsymbol{\xi}') | \boldsymbol{\xi}_j^s, \mathbf{u} \right\}, \quad (6.67)$$

kde $\tilde{V}^{s(i)}(\boldsymbol{\xi}') = \tilde{V}(\boldsymbol{\xi}')$ je hodnota aproximace optimální Bellmanovy funkce při znalosti hodnot $V_1^s, V_2^s, \dots, V_{N_\xi^s}^s$, $\boldsymbol{\xi}' = \boldsymbol{\phi}(\boldsymbol{\xi}_j^s, \mathbf{u}, \mathbf{x}')$ je hyperstav a \mathbf{x}' je spojitá část stavu v dalším časovém okamžiku. Počáteční hodnoty $V_j^{s(0)}$ pro $j = 1, 2, \dots, N_\xi^s$ mohou být zvoleny nulové

a celý postup může být ukončen následující zastavovací podmínkou

$$\|\tilde{V}^{s(i+1)} - \tilde{V}^{s(i)}\|_\infty \leq \epsilon_{VI}, \quad (6.68)$$

kde $\epsilon_{VI} > 0$ je mez přesnosti. Analogicky lze vypočítat hodnoty V_j^s pomocí dalších algoritmů dynamického programování jako je například iterace strategie.



Obrázek 6.9: Postup návrhu aktivního detektoru chyb s využitím neparametrické aproximace (6.65) Bellmanovy funkce.

Za předpokladu znalosti vzorků hyperstavu ξ_j^s a příslušných přibližných hodnot Bellmanovy funkce V_j^s , $j = 1, 2, \dots, N_\xi^s$ lze pomocí (6.65) aproximovat Bellmanovu funkci. Postup návrhu aktivního detektoru chyb s využitím neparametrické aproximace Bellmanovy funkce je ilustrován na obrázku 6.9. V prvním kroku je zvolena konečná diskretní množina hyperstavů \mathcal{G}^s , na základě které jsou vybraným iteračním algoritmem dynamického programování vypočítány přibližné hodnoty Bellmanovy funkce V_j^s , $j = 1, 2, \dots, N_\xi^s$. Tyto kroky jsou provedeny off-line. Následně je pro on-line použití navržen generátor vstupního signálu na základě neparametrické aproximace (6.65) Bellmanovy funkce. Celý postup je shrnut v algoritmu 8.

Algoritmus 8 Algoritmus využívající neparametrickou aproximaci získanou iterací uživatelské funkce pro úlohu aktivní detekce chyb.

1. Stanovení vzorků hyperstavů Volba množiny \mathcal{G}^s na základě zkušeností nebo simulací systému s různými podmínkami na počáteční hyperstav a vstupní signál.

2. Počáteční podmínky Inicializace Bellmanovy funkce $\bar{V}^{s(0)}$, volba meze přesnosti ϵ_{VI} a nastavení iteračního indexu $i = 0$.

3. Off-line: neparametrická aproximace Výpočet funkčních hodnot $V_1^s, \dots, V_{N_\xi}^s$ na základě iterace uživatelské funkce s krokem ohodnocení strategie (6.67) a ukončovací podmínky (6.68).

4. On-line: rozhodnutí a vstup Generování rozhodnutí d_k a vstupu \mathbf{u}_k v každém časovém okamžiku $k \in \mathcal{T}$ na základě strategie $\tilde{\rho} = [\tilde{\sigma}, (\tilde{\gamma})^T]^T$,

$$\tilde{\sigma}(\xi_k) = \arg \min_{d \in \mathcal{M}} \bar{L}^d(\xi_k, d), \quad (6.69)$$

$$\tilde{\gamma}(\xi_k) = \arg \min_{\mathbf{u} \in \mathcal{U}} E\{\tilde{V}(\xi_k) | \xi_k, \mathbf{u}\}, \quad (6.70)$$

kde \tilde{V} je dáno (6.65) a (6.66).

6.6 Dynamické programování pro úlohu aktivní detekce chyb a řízení

Předchozí část se týkala řešení úlohy aktivní detekce chyb bez požadavku na chování stavu systému. Řešení úlohy aktivní detekce chyb a řízení může být navrženo analogicky, je však důležité zohlednit všechny návrhové požadavky obecné ztrátové funkce. Z toho důvodu jsou v této části v krátkosti uvedena Bellmanova funkcionální rovnice pro úlohu zahrnující detekci chyb i řízení, a zároveň samotný předpis pro optimální aktivní detektor chyb a regulátor.

Nechť jsou uvažovány informační statistiky, které jsou poskládány do hyperstavu $\boldsymbol{\xi}_k \in \mathcal{G}$. Pro model chování (5.54) necht' je nalezena taková strategie (5.59), která minimalizuje kritérium (5.60). Bellmanova funkce popisující budoucí očekávané ztráty za předpokladu počátečního hyperstavu $\boldsymbol{\xi} \in \mathcal{G}$ a respektováním strategie $\bar{\rho}^r$ je definována jako

$$V^{\bar{\rho}^r}(\boldsymbol{\xi}) = \lim_{F \rightarrow \infty} \mathbb{E} \left\{ \sum_{k=0}^F \eta^k \bar{L}(\boldsymbol{\xi}_k, \bar{\sigma}(\boldsymbol{\xi}_k), \bar{\gamma}^r(\boldsymbol{\xi}_k)) | \boldsymbol{\xi}_0 = \boldsymbol{\xi} \right\}. \quad (6.71)$$

Podobně jako pro případ aktivní detekce chyb (6.24) může být v úloze aktivní detekce chyb a řízení Bellmanova funkcionální rovnice definována následovně

$$V^*(\boldsymbol{\xi}_k) = \min_{d_k \in \mathcal{M}, \mathbf{u}_k \in \mathcal{U}} \mathbb{E} \left\{ \bar{L}(\boldsymbol{\xi}_k, d_k, \mathbf{u}_k) + \eta V^*(\boldsymbol{\xi}_{k+1}) | \boldsymbol{\xi}_k, d_k, \mathbf{u}_k \right\}. \quad (6.72)$$

Stojí za připomenutí, že v případě úlohy aktivní detekce chyb je možné pravou stranu Bellmanovy funkcionální rovnice (6.25) rozdělit na dvě části. Jedna část popisuje minimální očekávané ztráty způsobené současným rozhodnutím o chybě a druhá část vyjadřuje celkové budoucí diskontované očekávané ztráty. Jelikož trajektorie hyperstavu pro úlohu aktivní detekce chyb a řízení závisí na vstupním řízení a ztrátová funkce závisí jak na vstupním řízení, tak na rozhodnutí, nelze rovnici (6.72) dále upravovat podobně jako (6.24).

Optimální strategie $\bar{\rho}^{r*}$ je určena řešením Bellmanovy funkcionální rovnice

$$\bar{\rho}^{r*}(\boldsymbol{\xi}_k) = \arg \min_{d_k \in \mathcal{M}, \mathbf{u}_k \in \mathcal{U}} \mathbb{E} \left\{ \bar{L}(\boldsymbol{\xi}_k, d_k, \mathbf{u}_k) + \eta V^*(\boldsymbol{\xi}_{k+1}) | \boldsymbol{\xi}_k, d_k, \mathbf{u}_k \right\}. \quad (6.73)$$

Dosazením předpisu (5.61) pro ztrátovou funkci do (6.73) a následnou úpravou je možné vyjádřit optimální generátor rozhodnutí a optimální regulátor jako

$$d_k^* = \bar{\sigma}^{r*}(\boldsymbol{\xi}_k) = \arg \min_{d_k \in \mathcal{M}} \alpha \bar{L}^d(d_k, \boldsymbol{\xi}_k), \quad (6.74)$$

$$\mathbf{u}_k^* = \bar{\gamma}^{r*}(\boldsymbol{\xi}_k) = \arg \min_{\mathbf{u}_k \in \mathcal{U}} \mathbb{E} \left\{ (1 - \alpha) \bar{L}^c(\boldsymbol{\xi}_k, \mathbf{u}_k) + \eta V^*(\boldsymbol{\xi}_{k+1}) | \boldsymbol{\xi}_k, \mathbf{u}_k \right\}. \quad (6.75)$$

Bellmanovu funkcionální rovnici (6.73) není možné pro formulovanou úlohu vyřešit analyticky a zároveň použití dynamického programování ztěžuje prokletí rozměrnosti v důsledku vlastností hyperstavu $\boldsymbol{\xi}_k$. K hledání suboptimálního řešení mohou být použity aproximační postupy, které byly prezentovány výše v této kapitole. Použití aproximačního dynamického programování a zpětnovazebního učení je analogické, jako v případě úlohy aktivní detekce chyb.

Kapitola 7

Detekce změn a chyb bez znalosti modelu

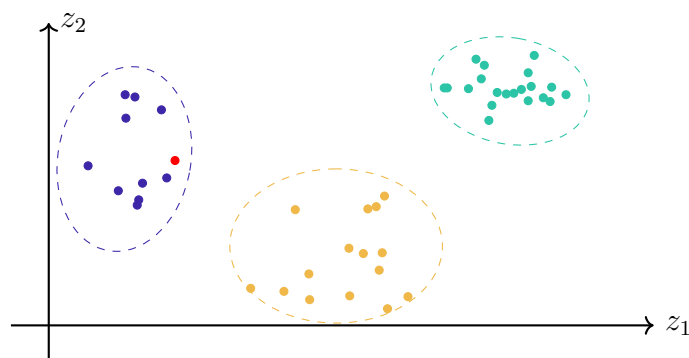
V této kapitole jsou navrženy dva adaptivní algoritmy detekce změn v systému bez znalosti modelu. Jejich úkolem je rozpoznat jednotlivá chování systému pouze na základě měřených vstupů a výstupů, za předpokladu, že nejsou k dispozici žádná již dříve klasifikovaná data. To znamená, že oba algoritmy sami vytvářejí návaznost mezi daty a možným chováním systému on-line v průběhu měření. První algoritmus prezentovaný v podkapitole 7.2 je odvozen pro lineární systémy a využívá vlastností lineárních zobrazení. Druhý algoritmus v podkapitole 7.3 využívá samoorganizující se mapy a jedná se o rozšíření algoritmu prezentovaného v [120]. Navržený algoritmus umožňuje porovnávat minulé modely chování s tím nově detekovaným, a tím je zamezeno zbytečnému přidávání nových modelů chování. Na rozdíl od předchozích částí se v této kapitole se hovoří hlavně o obecnější úloze detekce změn, která může být žádaná v úlohách automatického řízení.

7.1 Detekce změn a chyb založená na datech

Jak bylo uvedeno v kapitole 2, existuje celá řada směrů detekce změn a chyb v dynamických systémech. Převážná část této práce se zabývá návrhem aktivního detektoru chyb, který předpokládá úplnou znalost modelu systému. Vícemodelová detekce chyb tvoří významnou skupinu přístupů k detekce chyb, a to hlavně z důvodu detekce změn ve struktuře nebo parametrech. Samotná tvorba modelu rozsáhlých a komplexních systémů však může být časově i finančně náročná. Alternativní možností vytvoření modelu představuje identifikace systémů [121], kdy parametry modelu jsou identifikovány na základě dostupných dat. Detektor chyb v takovém případě sleduje změny parametrů modelu v čase. Někdy může být pevná struktura modelů limitující, a proto byly navrženy metody s pohyblivým počtem modelů [122]. Základní množina modelů je rozšířena nebo zúžena v závislosti na změně chování systému. Nicméně stěžejním bodem těchto technik je výběr samotné struktury modelu systému, která nemusí dostatečně charakterizovat chování systému.

Velká pozornost v oblasti detekce změn a chyb je věnována přístupu založenému na

datech, který se osvědčil v prostředí s velkým množstvím nasbíraných dat [2]. Inspirace metod tohoto přístupu sahá do strojového učení - jedné z mnoha podoblastí umělé inteligence [17]. Strojové učení lze rozdělit na učení s učitelem a učení bez učitele podle dostupnosti předem označených trénovacích dat. V případě učení s učitelem jsou dopředu poskytnuta trénovací data s jasnou návazností na chování systému. Příkladem metod založených na učení s učitelem jsou metody rozpoznání vzorů [43], které však mohou mít i varianty učení bez učitele. V případě rozpoznání vzorů proběhne klasifikace různých chování systému na základě dat nasbíraných ve zkušebním provozu a následně jsou vytvořeny hranice regionů chybného chování. V běžném provozu je pak na základě on-line dat rozpoznán vzor chování. Učení bez učitele představuje variantu, kdy není známa návaznost výstupních dat na vstupní data [123], a typickým příkladem je analýza hlavních komponent [124]. Analýza hlavních komponent se běžně využívá především k redukci dimenze dat, avšak poskytuje mimo jiné informaci o směru dat s největší variancí. Na základě této analýzy mohou být opět určeny hranice regionů běžného chování systému. Mezi další metody učení bez učitele patří metody založené na samoorganizujících se mapách [125], [126], rostoucí buněčné struktuře [127], nebo rostoucím neuronovém plynu [128]. Tyto metody dokáží rozdělit data do regionů, na jejichž základě lze analyzovat chování systému, jak je ilustrováno na obrázku 7.1. Výhodou metod založených na datech je, že nevyžadují znalost modelu.



Obrázek 7.1: Ilustrace přiřazení on-line měření (červeně) do příslušného regionu možných chování systému v případě dvou rozměrného vektoru měření $\mathbf{z} = [z_1, z_2]^T$.

Nechť je chování systému v každém časovém okamžiku $k \in \mathcal{T} = \{0, 1, 2, \dots\}$ popsáno modelem

$$\mathbf{x}_{k+1} = \mathbf{A}_{\mu_k} \mathbf{x}_k + \mathbf{B}_{\mu_k} \mathbf{u}_k, \quad (7.1)$$

kde $\mathbf{x}_k \in \mathbb{R}^{n_x}$ je spojitá část stavu, $\mu_k \in \mathcal{M} = \{1, 2, \dots, N_\mu\}$ je neznámá diskrétní část stavu, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ je vstup, \mathbf{A}_{μ_k} spolu s \mathbf{B}_{μ_k} jsou neznámé matice systému. Předpokládá se, že přepnutí modelu, tedy změna μ_k , nastane až po uplynutí dostatečně velkého časového intervalu tak, aby mohlo dojít k detekci změny. Chování systému je modelováno deterministickým stavovým modelem (7.1) s přímo měřitelnou spojitou částí stavu. Přestože se

jedná o jednoduchý model, který na rozdíl od předchozích kapitol neuvažuje nepřesnost modelování nebo vliv šumu na dynamiku systému, stále je velmi často využíván v úlohách řízení nebo detekce změn a chyb v systémech. Předpokládá se, že algoritmy detekce změny představených v této kapitole využívají vektorem měření $\mathbf{z}_k \in \mathbb{R}^{n_z}$, který obsahuje informaci o měřených datech. Diskuse výběru vhodného vektoru měření v kontextu samoorganizujících se map je uvedena například v [129], [130]. Necht je vektor měření zvolen následovně

$$\mathbf{z}_k^T = [\mathbf{x}_k^T, \mathbf{x}_{k-1}^T, \dots, \mathbf{x}_{k-N_o+1}^T, \mathbf{x}_{k-N_o}^T, \mathbf{u}_{k-1}^T, \dots, \mathbf{u}_{k-N_o+1}^T, \mathbf{u}_{k-N_o}^T]^T, \quad (7.2)$$

kde $N_o \in \mathbb{N}^+$ je počet po sobě jdoucích měření a $n_z = n_x(N_o + 1) + n_u N_o$ je dimenze vektoru měření. Cílem úlohy je správně detekovat změnu chování systému a rozpoznat, zda aktuální chování systému již dříve bylo detekováno.

7.2 Detekce změn využívající jádra lineárního zobrazení

Hlavní myšlenkou prvního algoritmu je využití geometrických vlastností lineárních systémů. Necht je pro ilustraci uvažován skalární lineární systém (7.1), jehož model může být upraven do následujícího tvaru

$$x_{k+1} - a_{\mu_k} x_k - b_{\mu_k} u_k = 0, \quad (7.3)$$

Za předpokladu tří dimenzionálního vektoru měření $\mathbf{z}_k = [z_{k,1}, z_{k,2}, z_{k,3}]^T = [x_k, x_{k-1}, u_{k-1}]^T$ může být vztah (7.3) upraven do vektorového tvaru

$$\bar{\mathbf{a}}_i^T \mathbf{z}_k = 0, \quad (7.4)$$

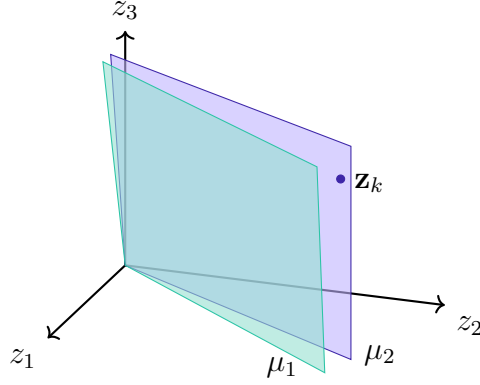
kde $\bar{\mathbf{a}}_i = [1, -a_i, -b_i]^T$ je vektor určující normálu roviny pro model $\mu_k = i$. Pak pro rozdílné μ_k leží vektor měření \mathbf{z}_k v rozdílné rovině, což je ilustrováno na obrázku 7.2. Změnu chování systému lze detekovat na základě ověření, zda vektor \mathbf{z}_k leží v jedné z rovin. V případě, že nenáleží ani jedné z nich, je zapotřebí určit novou rovinu charakterizující další možné chování systému. Úlohu je možné zobecnit i pro vyšší dimenzi n_z .

Model chování systému (7.1) je možné upravit jako

$$\mathbf{x}_{k+1} - \mathbf{A}_{\mu_k} \mathbf{x}_k - \mathbf{B}_{\mu_k} \mathbf{u}_k = \mathbf{0}. \quad (7.5)$$

Za předpokladu neměnného chování $\mu_k = i$ je možné na základě (7.5) a (7.2) sestavit následující vztah

$$\mathbf{M}_i \mathbf{z}_k = \mathbf{0}, \quad (7.6)$$



Obrázek 7.2: Ilustrace rovin možného výskytu měření $\mathbf{z}_k = [z_{k,1}, z_{k,2}, z_{k,3}]^T$ pro dvě možné chování systému. V tomto případě chování systému odpovídá indexu modelu $\mu_k = 2$.

kde $\mathbf{M}_i \in \mathbb{R}^{n_x N_o \times n_z}$ je matice definována jako

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{I} & -\mathbf{A}_i & \mathbf{0} & \dots & \mathbf{0} & -\mathbf{B}_i & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{A}_i & \mathbf{0} & \dots & \mathbf{0} & -\mathbf{B}_i & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & & \ddots & \ddots & & & & \ddots & & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{I} & -\mathbf{A}_i & \mathbf{0} & \dots & & \mathbf{0} & -\mathbf{B}_i \end{bmatrix}. \quad (7.7)$$

Jedná se o homogenní soustavu lineárních rovnic, kde \mathbf{z}_k je známý vektor a \mathbf{M}_i je neznámá matice. Aby bylo možné najít matici \mathbf{M}_i , je nutné získat $n_x + n_u N_o$ vektorů měření \mathbf{z}_k , přičemž první vektor měření může být zkonstruován v čase $k = N_o - 1$. Matice \mathbf{M}_i může být nalezena na základě řešení homogenní soustavy rovnic

$$\bar{\mathbf{Z}}_k^T \mathbf{M}_i^T = \mathbf{0}, \quad (7.8)$$

kde $\bar{\mathbf{Z}}_k \in \mathbb{R}^{n_z \times (n_x + n_u N_o)}$ je matice měření definovaná jako

$$\bar{\mathbf{Z}}_k = [\mathbf{z}_k, \mathbf{z}_{k+1}, \dots, \mathbf{z}_{k+n_x+n_u N_o-1}]. \quad (7.9)$$

Neznámá matice měření \mathbf{M}_i je nalezena jako nulový prostor matice $\bar{\mathbf{Z}}_k$, neboli

$$\mathbf{M}_i = \text{null}(\bar{\mathbf{Z}}_k^T)^T, \quad (7.10)$$

kde $\text{null}(\mathbf{D})$ je operátor, který vrací ortonormální bázi jádra lineárního zobrazení definovaného maticí \mathbf{D} získanou singulárním rozkladem argumentu [131].

Test změny chování systému spočívá ve výpočtu součinu vektoru měření se všemi známými maticemi \mathbf{M}_i , neboli

$$\mathbf{p}_k^i = \mathbf{M}_i \mathbf{z}_k, \quad i \in \mathcal{M}^a, \quad (7.11)$$

kde $\mathcal{M}^a = \{1, 2, \dots, N_\mu^a\}$ je aktuální množina všech indexů modelu, N_μ^a je aktuální počet možných chování a $\mathbf{p}_k^i \in \mathbb{R}^{n_x N_o}$ je signální vektor, který ovlivňuje výsledné hodnoty vah

jednotlivých modelů. V průběhu měření mohou být modely chování přidávány, přičemž aktuální počet modelů je indikován číslem $N_\mu^a \leq N_\mu$. Pokud se chování systému nezměnilo, pak musí platit rovnice (7.8). To znamená, že vektor \mathbf{p}_k^i musí být nulový pro i , které odpovídá aktuálnímu indexu modelu μ_k . Stanovením meze tolerance $\epsilon_n > 0$ a porovnáním s normou vektoru $\|\mathbf{p}_k^i\|_2$ lze rozhodnout, zda došlo ke změně chování systému, nebo zda je odchylka způsobena nepřesností při numerickém výpočtu. V případě, že hodnota $\|\mathbf{p}_k^i\|_2$ překročí ϵ_n pro všechna $i \in \mathcal{M}^a$, musí být přidán nový model chování, neboli $N_\mu^a = N_\mu^a + 1$. V opačném případě jsou vypočítány váhy modelů jako

$$m_k^i = \frac{1}{\sum_{j=1}^{N_\mu^a} \frac{\|\mathbf{p}_k^j\|_2}{\|\mathbf{p}_k^i\|_2}}, \quad i \in \mathcal{M}^a, \quad (7.12)$$

kde $m_k^i \in \langle 0, 1 \rangle$ je váha modelu $\mu_k = i$ v čase $k \in \mathcal{T}$, $\sum_{i=1}^{N_\mu^a} m_k^i = 1$. Pro $\|\mathbf{p}_k^i\|_2 = 0$ je váha i -tého modelu nastavena na hodnotu jedna a ostatní váhy modelů jsou nulové, neboli $m_k^i = 1$, $m_k^l = 0$, $l \in \mathcal{M}^a$, $l \neq i$. Smysl vah m_k^i je podobný tomu uvedenému v kapitole 5, kdy index modelu μ_k byl nahrazen podmíněnými pravděpodobnostmi $P(\mu_k = i | \mathbf{Y}_0^{k+1})$ pro všechna $i \in \mathcal{M}$.

Postup detekce změny chování systému založený na myšlence nulového prostoru je shrnutý v algoritmu 9. Algoritmus na začátku předpokládá pouze jedno možné chování systému, které se v prvních $N_o + n_x + n_u N_o - 1$ časových okamžicích nemění. Po sestavení matice $\bar{\mathbf{Z}}_0$ z počátečních měření je nejprve vypočítána matice \mathbf{M}_1 pro počáteční chování systému označené indexem modelu $\mu_k = 1$. V následujícím časovém okamžiku je sestaven aktuální vektor měření. Je nutné poznamenat, že během počátečního měření a po každé změně chování je předpokládáno, že chování systému zůstane stejné do okamžiku, než je možné určit novou matici \mathbf{M}_{μ_k} . Aktuální chování systému pak odpovídá případu $\mu_k = i$, pro které platí $m_k^i \geq \frac{1}{|\mathcal{M}^a|}$, kde $|\mathcal{M}^a|$ značí mohutnost množiny \mathcal{M}^a .

7.3 Detekce změn využívající samoorganizující se mapy

Samoorganizující se mapy jsou rozšířeným a univerzálním nástrojem sloužícím k analýze neznámého signálu [126]. Jedná se o samoučící se neuronové sítě, které mohou sloužit k rozdělení nějaké množiny vzorů na skupiny se stejnými nebo podobnými rysy. Z hlediska časově proměnného chování systému to znamená, že samoorganizující se mapy mohou na základě naměřených vstupně výstupních dat rozpoznat různá chování systému.

Samoorganizující se mapy se skládají ze dvou vrstev: vstupní a výpočetní. Vstupní vrstva je tvořena vstupním vektorem, zatímco výpočetní vrstva se skládá z neuronů a jejich vah. Na základě rozdílnosti vstupního vektoru a vah jednotlivých neuronů je určen vítězný neuron, který nejvíce odpovídá vstupnímu signálu. Na neuron může být pohlíženo jako na index modelu chování systému. Necht $\boldsymbol{\omega}_k^i = [\omega_k^{i,1}, \omega_k^{i,2}, \dots, \omega_k^{i,n_\omega}]^T \in \mathbb{R}^{n_\omega}$ je váhový vektor neuronu $i \in \mathcal{M} = \{1, 2, \dots, N_\mu\}$, kde $\omega_k^{i,j} \in \mathbb{R}$ je váha spojující j -tý prvek vstupního vektoru měření \mathbf{z}_k a neuronu i v časovém okamžiku k . Samoorganizující se mapa určí

Algoritmus 9 Adaptivní algoritmus využívající vlastností nulového prostoru pro detekci změn chování bez znalosti modelu.

Inicializace Nastavení počtu známých měření N_o , aktuálního počtu modelů $N_\mu^a = 1$ a tolerance ϵ_n .

1. Sběr dat Sestavení vektorů měření podle (7.2). V případě indikace změny modelu nebo při inicializaci je nutné sestavit matici \mathbf{M}_i na základě (7.10).

2. Test změny chování Zkoumání změny chování může být provedeno jedině tehdy, je-li známá matice \mathbf{M}_i všechna $i \in \{1, 2, \dots, N_\mu^a\} = \mathcal{M}^a$.

2.1) Výpočet \mathbf{p}_k^i pro všechna $i \in \mathcal{M}^a$ podle (7.11).

2.2) Jestliže $\|\mathbf{p}_k^i\|_2 > \epsilon_n$ pro všechna $i \in \mathcal{M}^a$, je detekována změna chování a musí být přidán nový model chování, $N_\mu^a = N_\mu^a + 1$, $m_k^{N_\mu^a} = 1$ a $m_k^i = 0$ pro $i \in \{1, 2, \dots, N_\mu^a - 1\}$. V opačném případě jsou vypočítány váhy modelů systému na základě (7.12).

Nastavení $k = k + 1$ a skok zpět na krok 1.

vítězný vektor vah $\boldsymbol{\varpi}_k^{i*} \in \mathbb{R}^{n_\varpi}$ jako vektor odpovídající vítěznému neuronu

$$i_k^* = \arg \min_{i \in \mathcal{M}} \|\mathbf{z}_k - \boldsymbol{\varpi}_k^i\|_2. \quad (7.13)$$

Váhy vítězného neuronu a všechny váhy v jeho okolí jsou upraveny směrem k hodnotám vstupního vektoru \mathbf{z}_k využitím následujícího pravidla

$$\boldsymbol{\varpi}_{k+1}^i = \boldsymbol{\varpi}_k^i + \kappa_k \varrho(\boldsymbol{\varpi}_k^{i*}, \boldsymbol{\varpi}_k^i)(\mathbf{z}_k - \boldsymbol{\varpi}_k^i), \quad (7.14)$$

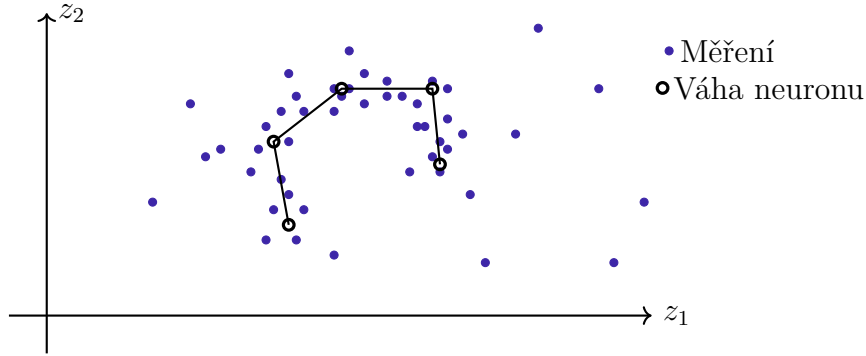
kde $\kappa_k \geq 0$ je parametr rychlosti učení a $\varrho : \mathbb{R}^{n_\varpi} \times \mathbb{R}^{n_\varpi} \mapsto \mathbb{R}$ je funkce okolí, typicky definovaná jako

$$\varrho(\boldsymbol{\varpi}_k^{i*}, \boldsymbol{\varpi}_k^i) = \exp\left(\frac{\|\boldsymbol{\varpi}_k^i - \boldsymbol{\varpi}_k^{i*}\|_2}{2\varsigma_k^2}\right), \quad (7.15)$$

kde $\varsigma_k > 0$ je parametr určující šířku okolí. Parametry κ_k a ς_k jsou časově proměnné a nejlepšího chování samoorganizující se mapy je dosaženo, pokud hodnoty parametrů κ_k a ς_k monotónně klesají s časem k nule. V důsledku totiž rychlost učení a velikost dotčeného okolí vítězného neuronu s časem také klesá. Požadovaných vlastností může být dosaženo pokud jsou parametry nastaveny tak, že

$$\kappa_k = \kappa_0 \exp\left(-\frac{k}{\tau_\kappa}\right), \quad \varsigma_k = \varsigma_0 \exp\left(-\frac{k}{\tau_\varsigma}\right), \quad (7.16)$$

kde $\kappa_0 > 0$ spolu s $\varsigma_0 > 0$ jsou počáteční hodnoty příslušných parametrů a $\tau_\kappa > 0$ spolu s $\tau_\varsigma > 0$ jsou rychlosti zapomínání odpovídajících parametrů. Rozložení pevného počtu pěti vah $\boldsymbol{\varpi}_k^i$ samoorganizující se mapy na základě dostupných měření pro případ $n_z = 2$



Obrázek 7.3: Ilustrace samoorganizující se mapy podle rozložení dostupných měření.

je ilustrováno na obrázku 7.3. Z obrázku je patrné, že váhy kopírují tvar oblasti největšího výskytu měření.

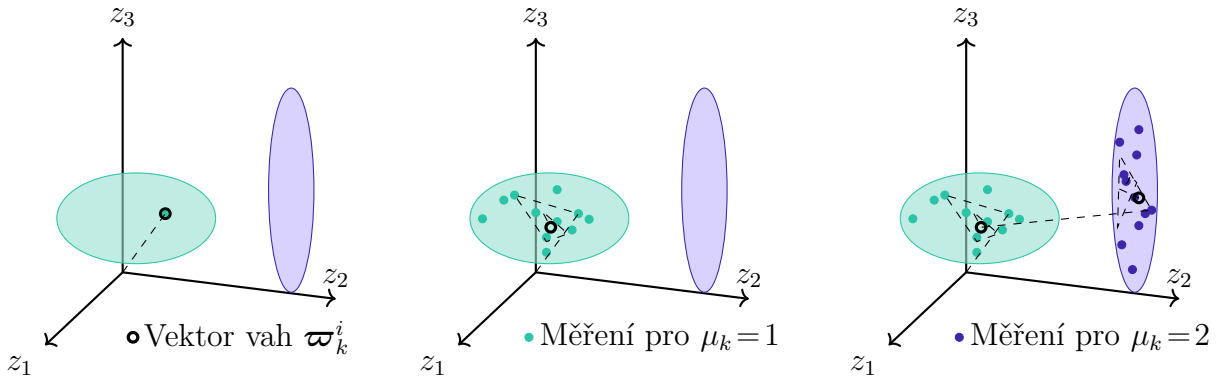
V běžném použití samoorganizujících se map se předpokládá, že výpočty jsou vykonávány nad celou množinou možných chování systému. Avšak v případě časově proměnných systémů odpovídají vstupní naměřená data počátečnímu chování systému, které se s časem může změnit. K takové změně může dojít opakovaně. Z toho důvodu je zapotřebí vytvořit adaptivní algoritmus samoorganizujících se map, který umožní začít pouze s jedním možným chováním systému a v závislosti na vstupně-výstupních datech ovlivněných změnami chování systému adaptovat počet možných modelů.

Nechť se předpokládá, že v každém časovém okamžiku je chování systému popsáno právě N_μ^a modely, $N_\mu^a \leq N_\mu$, kde N_μ^a je adaptivní počet modelů. Nový model chování systému je přidán, jestliže vstupní vektor měření \mathbf{z}_k neodpovídá žádnému z dosud poznatých chování. Jednotlivé váhy modelů m_k^i , určující míru závislosti vektoru měření \mathbf{z}_k na modelu chování, tedy neuronu i , mohou být definovány jako

$$m_k^i = \frac{\exp\left(-\frac{\|\mathbf{z}_k - \boldsymbol{\varpi}_k^i\|_2}{2\beta}\right)}{\sum_{j=1}^{N_\mu^a} \exp\left(-\frac{\|\mathbf{z}_k - \boldsymbol{\varpi}_k^j\|_2}{2\beta}\right)}, \quad i = 1, 2, \dots, N_\mu^a, \quad (7.17)$$

kde $m_k^i \in \langle 0, 1 \rangle$ je váha modelu $\mu_k = i$, $\sum_{i=1}^{N_\mu^a} m_k^i = 1$ a $\beta > 0$ je parametr, který měří míru rozsahu hodnot \mathbf{z}_k kolem $\boldsymbol{\varpi}_k^i$. V důsledku (7.17) jsou všechny váhy modelů stejné pro velké hodnoty β a naopak.

Vývoj vektoru vah $\boldsymbol{\varpi}_k^i$ navrženého adaptivního algoritmu samoorganizujících se map v závislosti na hodnotách vektoru měření \mathbf{z}_k pro $n_z = 3$ a $N_\mu = 2$ je ilustrován na obrázku 7.4. Vybarvené plochy představují množiny možného výskytu \mathbf{z}_k pro index modelu $\mu_k \in \mathcal{M}$. Obrázek vlevo představuje počáteční nastavení vah $\boldsymbol{\varpi}_{k_0}^1$ na hodnoty \mathbf{z}_{k_0} v časovém okamžiku $k_0 \in \mathcal{T}$, kdy je nasbíráno dostatečné množství dat na sestavení vektoru měření. Po několika měřeních se vektor vah $\boldsymbol{\varpi}_k^1$ ustálí, což ilustruje obrázek uprostřed. Vývoj vektoru vah závisí na získaných měření. Při následné změně chování systému je přidán nový vektor vah $\boldsymbol{\varpi}_k^2$, jehož hodnoty se opět naladí v závislosti na dostupných měřeních, jak je ukázáno na obrázku vpravo. Důležitým krokem adaptivního algoritmu je testování



Obrázek 7.4: Ilustrace vývoje vektoru vah ϖ_k^i navrženého adaptivního algoritmu v závislosti na hodnotách vektoru měření \mathbf{z}_k .

změny chování, které může být provedeno testováním překročení předem stanovené meze tolerance ϵ_d hodnotou Euklidovy normy rozdílu mezi ϖ_k^i a \mathbf{z}_k . Adaptivní postup detekce změn chování využívající samoorganizující se mapy je uveden v algoritmu 10. Mezi slabé stránky algoritmu patří nutná volba celé řady parametrů, které mohou zásadním způsobem ovlivnit chování algoritmu. V případě vyšší dimenze vektoru měření \mathbf{z}_k se volba parametrů stává nepřehledná. Navíc v případě blízkých regionů výskytu možných trajektorií \mathbf{z}_k pro různé chování systému nemusí detekce založená výhradně na porovnávání hodnot $\|\varpi_k^i - \mathbf{z}_k\|_2$ a ϵ_d fungovat, jelikož adaptivní algoritmus předpokládá jeden vektor vah pro jedno chování systému. Naopak výhodou algoritmu je možné použití také pro nelineární systémy.

7.4 Využití detekce změn v úloze optimálního řízení bez znalosti modelu

Optimální regulátory cílí na minimalizaci zvoleného návrhového kritéria hodnotícího kvalitu řízení. Velmi známou úlohu představuje úloha LQR [132]. Optimální regulátor je navržen off-line řešením Riccatiho rovnice s využitím úplné znalosti dynamiky systému, tedy matice dynamiky a matice vstupu. V případě, že dynamika systému není známa, je možné využít zpětnovazební učení. Návrh regulátoru probíhá on-line na základě naměřených dat.

Velká část metod zpětnovazebního učení je založena na struktuře *herec-kritik* [93]. Herec v podobě strategie řízení generuje vstup do systému a kritik tento vstupní signál ohodnotí z hlediska návrhových požadavků. Na základě ohodnocení je herec aktualizován ve smyslu vylepšení strategie generující řídicí vstup. Herec a kritik mohou být reprezentovány lineární parametrickou aproximací, která byla diskutována v podkapitole 6.4.1. Návrh optimálního regulátoru se strukturou herec-kritik byl diskutován například v [133]. Další možnost návrhu regulátoru bez znalosti modelu představuje zpětnovazební učení využívající Q-funkci.

Algoritmus 10 Adaptivní algoritmus samoorganizujících se map pro detekci změn chování bez znalosti modelu.

Inicializace Nastavení počtu modelů $N_\mu^a = 1$, parametrů $\beta, \kappa_0, \varsigma_0, \tau_\kappa, \tau_\varsigma$ a meze detekce nového chování $\epsilon_d > 0$. Nastavení $\bar{k}_{k_0} = 0$ a inicializace $\varpi_{k_0}^1 = \mathbf{z}_{k_0}$.

Pro každý časový okamžik $k \geq k_0$:

1. Konstrukce vektoru měření Sestavení \mathbf{z}_k z měřených dat podle (7.2).

2. Test změny chování Testování náhlé změny chování je následující.

2.1) Jestliže platí $\|\varpi_k^i - \mathbf{z}_k\|_2 > \epsilon_d$ pro všechny $i \in \{1, 2, \dots, N_\mu^a\}$, pak je nutné provést další měření tak, aby vektor \mathbf{z}_k byl naplněn pouze novými měřeními. Po dokončení sběru dat je znovu proveden test shody. Pokud opět platí podmínka, je přidán nový model $N_\mu^a = N_\mu^a + 1$ a následuje nastavení hodnot $\varpi_k^{N_\mu^a} = \mathbf{z}_k$ a $\bar{k}_k = 0$

2.2) Pokud platí $\|\varpi_k^i - \mathbf{z}_k\|_2 \leq \epsilon_d$ pro jakékoli $i \in \{1, 2, \dots, N_\mu^a\}$, nový model přidán není.

3. Nastavení parametrů Nastavení hodnot parametrů v daném pořadí

$$\kappa_k = \kappa_0 \exp\left(-\frac{\bar{k}_k}{\tau_\kappa}\right), \quad \varsigma_k = \varsigma_0 \exp\left(-\frac{\bar{k}_k}{\tau_\varsigma}\right), \quad \bar{k}_k = \bar{k}_k + 1. \quad (7.18)$$

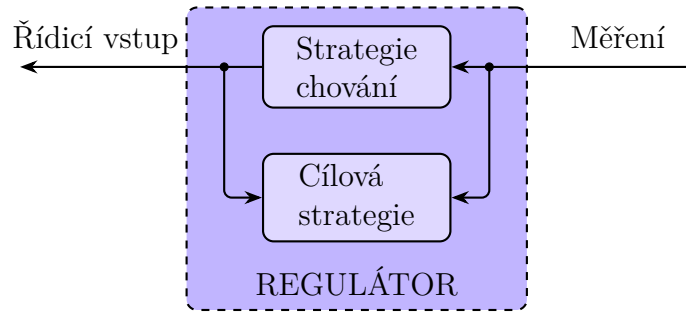
4. Vylepšení vážení modelů Výpočet m_k^i pro všechny neurony $i \in \{1, 2, \dots, N_\mu^a\}$ podle (7.17).

5. Vylepšení vah neuronů Výpočet nového vektoru vah ϖ_{k+1}^i pro všechny $i \in \{1, 2, \dots, N_\mu^a\}$ podle (7.14).

Skok zpět na krok 1 dokud není splněna ukončovací podmínka.

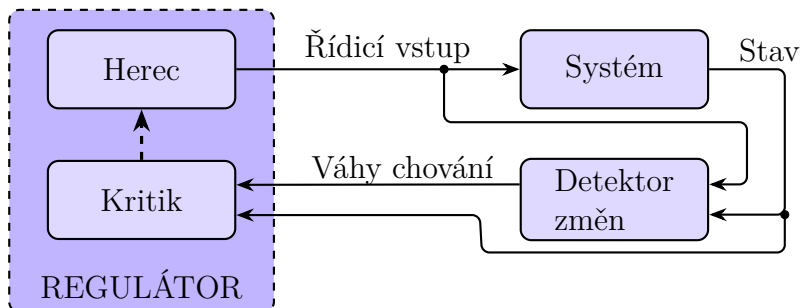
Na rozdíl od struktury herec-kritik nevyžaduje Q-učení úplnou znalost dynamiky systému. Návrh regulátoru pro sledování referenčního signálu pomocí Q-učení je uveden v [134]. Jednu z technik, která v poslední době získala zvýšenou pozornost, představuje tzv. off-policy učení. *Off-policy učení* je založené na trénování tzv. *cílové strategie*, neboli hledané strategie řízení, zatímco je řídicí vstupní signál systému generovaný jinou strategií, která je označována jako *strategie chování*. Vnitřní schéma regulátoru navrženého pomocí off-policy učení je ilustrováno na obrázku 7.5. Off-policy učení, stejně jako Q-učení, nevyžaduje žádnou znalost dynamiky systému. Návrh optimálního řízení pomocí off-policy učení je pro lineární diskrétní systémy poprvé diskutováno v [135], avšak v případě spojitých systémů je off-policy učení již rozšířeno několik let [136]–[138].

Náročnou úlohu představuje návrh optimálního regulátoru pro časově proměnné systémy. V některých úlohách bez znalosti modelu chování systému nemusí být zpětnovazební učení dostatečně flexibilní a regulátor nemusí rychle reagovat na náhlou změnu chování systému. V takových případech najdou uplatnění algoritmy detekce změn. Návrh regulátoru využívající strukturu herec-kritik a algoritmus detekce změn založený na samoorga-



Obrázek 7.5: Vnitřní struktura regulátoru navrženého off-policy učením.

nizujících se mapách je ukázán v [120]. Právě tento algoritmus detekce změn byl rozšířen v podkapitole 7.3 o možnost přenastavení parametrů κ_k a ς_k ovlivňující rychlost učení a velikost okolí vítězného neuronu. Zároveň je navržený algoritmus schopný testovat podobnost současného chování s tím předchozím, čímž je zabráněno nekontrolovanému přidávání nových modelů chování. V případě lineárních systémů je možné využít algoritmus navržený v podkapitole 7.2, který lépe využívá vlastností lineárních systémů. Základní blokové schéma použití detektoru změn v případě struktury regulátoru herec-kritik je uvedeno na obrázku 7.6. Detektor změn na základě dostupných dat určí váhy možných chování systému. Na základě těchto vah chování je vylepšen regulátor, který generuje řídicí vstup do systému. Detailní návrh optimálního regulátoru se strukturou herec-kritik pomocí off-policy učení a algoritmu 10 byl autorem publikován v [Š1].



Obrázek 7.6: Vnitřní struktura regulátoru navrženého off-policy učením.

Kapitola 8

Ilustrační příklady

Tato kapitola je věnována ilustraci a simulačnímu ověření navržených algoritmů na příkladech, které zahrnují jak jednoduché ilustrační modely, tak modely složitějších reálných systémů jako je chemostat nebo mobilní robot. Podkapitola 8.1 je zaměřena na úlohu aktivní detekce chyb a řízení v dynamických systémech, zatímco podkapitola 8.2 se věnuje detekci změn na základě dat bez znalosti modelu. Všechny uvedené úlohy jsou řešeny v simulačním prostředí MATLAB R2016a na počítači s procesorem Intel Core i5-3570 CPU @ 3.40GHz a pamětí 8GB RAM.

8.1 Aktivní detekce chyb a řízení

8.1.1 Chemostat

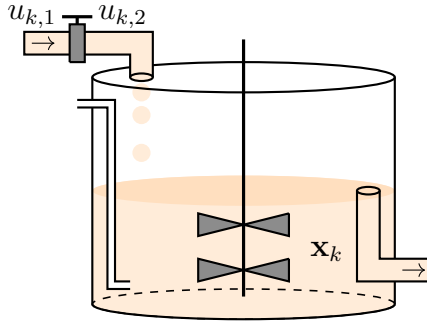
První příklad je zaměřen na ověření návrhu aktivního detektoru chyb a regulátoru pro modely s přímo měřitelnou spojitou částí stavu. Návrh řešení je založen na přeformulování úlohy na úlohu s úplnou informací o stavu pomocí odpovídajících informačních statistik uvedené v podkapitole 5.2.1. K samotnému řešení optimalizační úlohy se využije adaptivní algoritmus 4 zobecněné iterace strategie, který je založený na transformaci modelu na Markovův rozhodovací proces. Adaptivní algoritmus je porovnán dalšími algoritmy dynamického programování.

Chemostat je kontinuálně míchaný bioreaktor, který může sloužit například ke kvašení rostlinné buněčné kultury [139]. Substrát, který je zapotřebí k růstu bakterií, je neustále přidáván do chemostatu. Směs substrátu a bakterií se pak odvádí z nádoby odtokem. Zároveň se předpokládá, že v chemostatu dochází k důkladnému míchání obsahu nádoby. Schéma chemostatu je ilustrováno na obrázku 8.1. Použitím dopředné Eulerovy metody s periodou vzorkování $T_s = 1$ [hr] může být dynamické chování systému v každém časovém okamžiku $k = 0, 1, \dots$ popsáno následujícím nelineárním diskretním stochastickým

stavovým modelem

$$\begin{aligned} x_{k+1,1} &= x_{k,1} + \frac{T_s c_1 x_{k,1} x_{k,2}}{c_2 + x_{k,2}} - T_s u_{k,1} x_{k,1} + w_{k,1}, \\ x_{k+1,2} &= x_{k,2} - \frac{T_s c_1 x_{k,1} x_{k,2}}{c_3 (c_2 + x_{k,2})} + T_s u_{k,1} (u_{k,2} - x_{k,1}) + w_{k,2}, \end{aligned} \quad (8.1)$$

kde $\mathbf{x}_k = [x_{k,1}, x_{k,2}]^T \in \mathbb{R}^2$ je spojitá část stavu systému, $x_{k,1}$ je koncentrace bakterií, $x_{k,2}$ je koncentrace substrátu, $\mathbf{u}_k = [u_{k,1}, u_{k,2}]^T \in \mathbb{R}^2$ je vstup, $u_{k,1}$ je průtok substrátu podělený objemem chemostatu, $u_{k,2}$ je koncentrace substrátu a $\mathbf{w}_k = [w_{k,1}, w_{k,2}]^T \in \mathbb{R}^2$ jsou nezávislé stavové šумы definované Laplaceovým rozdělením pravděpodobnosti s nulovou střední hodnotou $\mathbf{m}^w = [0, 0]^T$ a kovariancí $\Sigma^w = 0.288 \cdot 10^{-5} \mathbf{I}_2$. Parametry c_1 , c_2 a c_3 představují v tomto pořadí maximální rychlost růstu bakterií, množství substrátu rozpuštěné při poloviční specifické rychlosti využití substrátu a parametr výnosu. Předpokládají se dvě možné chování systému. V případě bezchybného chování systému jsou hodnoty parametrů $c_1 = 0.2$ [hr^{-1}], $c_2 = 1$ [$\text{g} \cdot \text{l}^{-1}$], $c_3 = 0.5$ [$\text{g} \cdot \text{g}^{-1}$], v případě chybného chování se hodnota parametru c_2 nepatrně změnila na $c_2 = 0.9$ [$\text{g} \cdot \text{l}^{-1}$]. Přechod mezi bezchybným a chybným modelem je definován pomocí pravděpodobnostní přechodů $P(\mu_{k+1} = j | \mu_k = i) = 0.02$ pro $i, j \in \mathcal{M}, i \neq j$. Při návrhu aktivního detektoru chyb a regulátoru jsou uvažovány následující návrhové požadavky. Cílem detekce je minimalizace pravděpodobnosti chybného rozhodnutí, zatímco cílem řízení je minimalizace ztrát spojených s generováním nenulového vstupu a zároveň sledování konstantního referenčního signálu $\mathbf{r}_k = [r_{k,1}, r_{k,2}]^T = [4.4, 1]^T$ spojitou částí stavu \mathbf{x}_k . Množina přípustných řízení je definována jako $\mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2$, $\mathcal{U}_1 = \{0, 0.05, 0.1\}$, $\mathcal{U}_2 = \{0, 1, \dots, 10\}$.



Obrázek 8.1: Ilustrační schéma chemostatu.

Spojitý hyperstav $\zeta_k = [\zeta_{k,1}, \zeta_{k,2}, \zeta_{k,3}]^T$ je definován jako $\zeta_k = [\mathbf{x}_k^T, P(\mu_k = 1 | \mathbf{Y}_0^k)]^T \in \mathcal{R}$, $\mathbf{Y}_0^k = [(\mathbf{x}_0^k)^T, (\mathbf{u}_0^{k-1})^T]^T$. Necht je nad vybranou oblastí spojitého prostoru hyperstavů vytvořena rovnoměrná mřížka $\mathcal{R}^g = \mathcal{R}_1^g \times \mathcal{R}_2^g \times \mathcal{R}_3^g$, která je definovaná množinami $\mathcal{R}_1^g = \{0, 0.05, \dots, 4.5\}$, $\mathcal{R}_2^g = \{0, 0.05, \dots, 2.5\}$ a $\mathcal{R}_3^g = \{0, 0.01, \dots, 1\}$. Detailní postup návrhu diskrétní mřížky spolu s přeformulováním úlohy na Markovův rozhodovací proces je uveden v podkapitole 6.2. Podmíněné pravděpodobnosti přechodů mezi hyperstavů jsou vypočítány podle vztahu (6.4) pomocí 500 MC simulací. Ztrátová funkce má následující strukturu

$$L(\zeta_k, d_k, \mathbf{u}_k) = \alpha L^d(\zeta_k, d_k) + (1 - \alpha) L^c(\zeta_k, \mathbf{u}_k), \quad (8.2)$$

kde $\alpha = 0.5$ je váhový koeficient a dílčí ztrátové funkce detekce L^d a řízení L^c jsou zvoleny následovně

$$L^d(\boldsymbol{\zeta}_k, d_k) = \begin{cases} 1 - \zeta_{k,3} & \text{if } d_k = 1, \\ \zeta_{k,3} & \text{if } d_k = 2, \end{cases} \quad (8.3)$$

$$L^c(\boldsymbol{\zeta}_k, \mathbf{u}_k) = \sum_{i=1}^2 |p_i^u u_{k,i}| + \sum_{i=1}^2 p_i^\zeta \left(1 - \exp\left(-q_i(\zeta_{k,i} - r_{k,i})^2\right)\right), \quad (8.4)$$

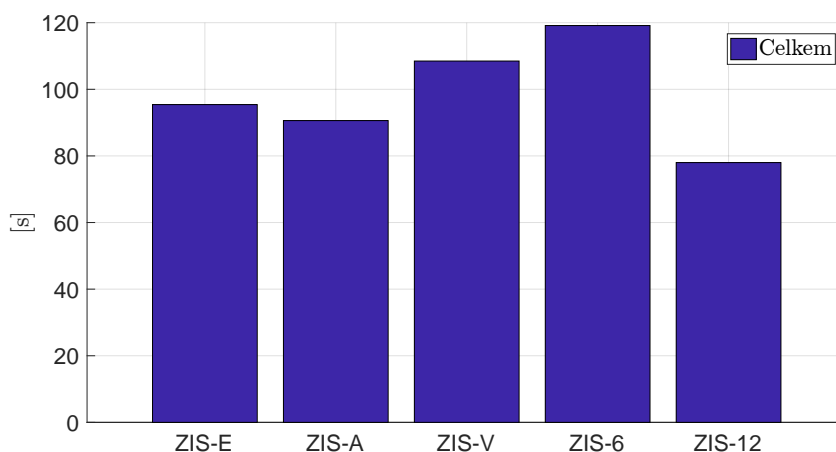
kde parametry uživatelské funkce jsou definovány následovně $\mathbf{p}^u = [p_1^u, p_2^u]^T = [0.01, 0.03]^T$, $\mathbf{p}^\zeta = [p_1^\zeta, p_2^\zeta]^T = [5, 1]^T$, $\mathbf{q} = [q_1, q_2]^T = [1, 1]^T$. Diskontní faktor je zvolen jako $\eta = 0.91$.

Celkem je diskrétní mřížka tvořena 468741 diskrétními hyperstavy. Pravděpodobnosti přechodů mezi hyperstavy jsou proto uloženy ve formě řídké matice tak, aby se snížily vysoké paměťové požadavky. V důsledku jsou všechny výpočty realizovány pomocí specializovaných algoritmů využívajících vlastnosti řídkých matic. Použití těchto algoritmů zároveň snižuje výpočetní nároky.

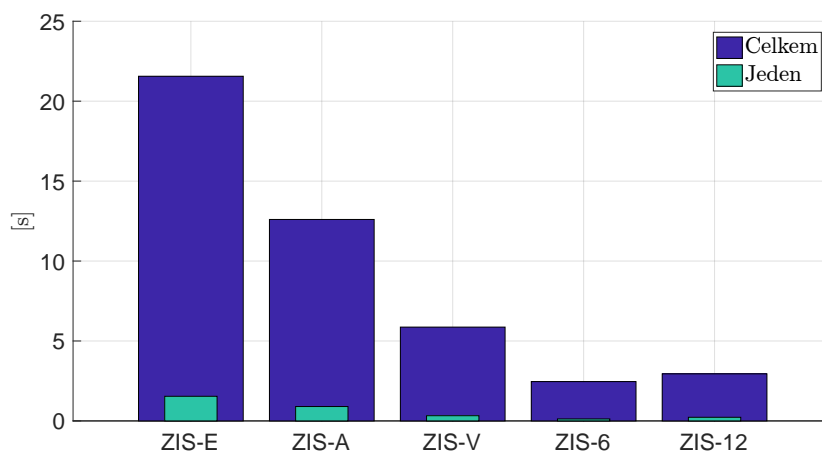
Vlastnosti algoritmu 4 zobecněné iterace strategie (ZIS-A), který určuje hodnotu \bar{j} počtu iterací v kroku ohodnocení strategie adaptivně, jsou srovnány s následujícími algoritmy. Krok ohodnocení strategie je v algoritmu zobecněné iterace strategie (ZIS-E) proveden řešením systému lineárních rovnic pomocí operátoru zpětného lomítka využívající sadu rutin UMFPAK. Algoritmus ZIS-E tedy odpovídá iteraci strategie, neboli $\bar{j} \rightarrow \infty$. Další dva algoritmy ZIS-6 a ZIS-12 používají v pořadí $\bar{j} = 6$ a $\bar{j} = 12$ iterací k nalezení řešení kroku ohodnocení strategie. Poslední algoritmus ZIS-V využívá pevného předpisu $\bar{j} = 2 + 2i$, kde $i = 0, 1, 2, \dots$ je iterace algoritmu zobecněné iterace strategie. Mez přesnosti postupné aproximace algoritmu ZIS-A je zvolena jako $\epsilon_{SA} = 0.5$ a maximální počet iterací je omezen $\bar{j} < 110$ na základě simulačních experimentů. Porovnání algoritmů ZIS-A a ZIS-E je smysluplné, jelikož v případě neiteračního řešení systému lineárních rovnic v kroku ohodnocení strategie se využívají stejné techniky. Vnější iterační cyklus zobecněné strategie řízení je ukončen, pokud je splněna jedna z následujících podmínek:

- shoda dvou po sobě jdoucích strategií $\bar{\gamma}^{r(i+1)} = \bar{\gamma}^{r(i)}$ generující řídicí vstup,
- hodnoty Bellmanovy funkce se s přibývajícím iteracemi nemění $\|\bar{V}_{\bar{\rho}^{r(i+1)}} - \bar{V}_{\bar{\rho}^{r(i)}}\|_\infty \leq 10^{-5}$ a zároveň bylo dosaženo minimálního počtu iterací $i \geq 30$,
- dosažení maximálního počtu iterací $i = 3000$.

Všechny algoritmy zobecněné iterace strategie byly ukončeny podmínkou shody dvou po sobě jdoucích strategií, avšak s rozdílným počtem vnějších iterací. Zatímco ZIS-E a ZIS-A potřebovaly k nalezení řešení 14 iterací a ZIS-12 dokonce jen 13 iterací, ZIS-V a ZIS-6 potřebovaly v pořadí 18 a 20 iterací. Srovnání celkových obvyklých výpočetních časů na vykonání celé zobecněné iterace strategie pro všechny algoritmy je zobrazeno na obrázku 8.2. Z výsledků je patrné, že celkově nejmenší výpočetní čas byl vyžadován algoritmem ZIS-12. Z ostatních algoritmů si nejlépe vedl navržený adaptivní algoritmus ZIS-A. Naopak nejvíce

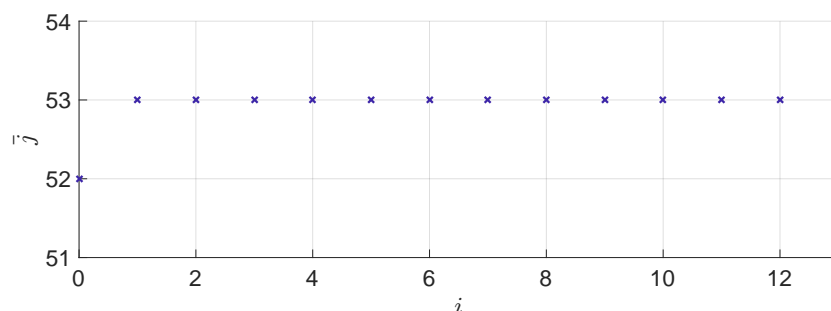


Obrázek 8.2: Srovnání celkových obvyklých výpočetních časů na vykonání celé zobecněné iterace strategie včetně kroku ohodnocení strategie pro všechny srovnávané algoritmy pro systém chemostatu.



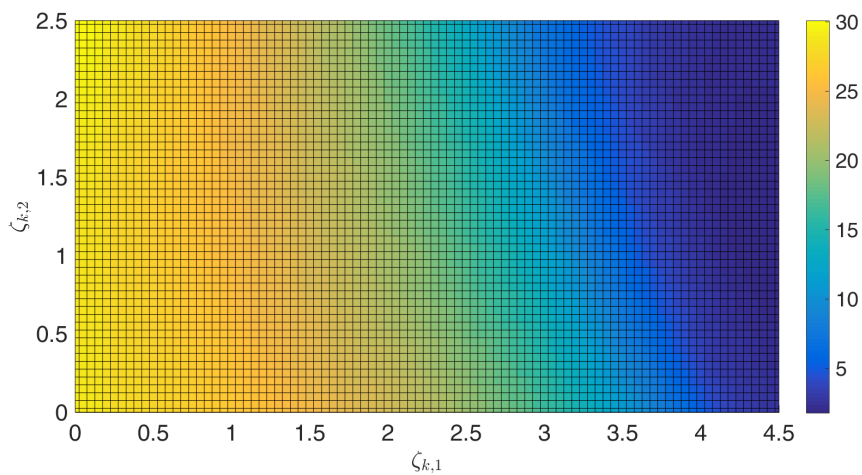
Obrázek 8.3: Srovnání obvyklých výpočetních časů na vykonání jednoho kroku ohodnocení strategie a celkového času na vykonání ohodnocení strategie pro všechny srovnávané algoritmy pro systém chemostatu.

času bylo zapotřebí k nalezení řešení pomocí ZIS-6. Všechny výpočetní časy jsou ovlivněny jak celkovým počtem iterací zobecněné iterace strategie, tak výpočetními časy kroku ohodnocení strategie. Srovnání obvyklých výpočetních časů na vykonání jednoho kroku ohodnocení strategie a celkového času na vykonání ohodnocení strategie jsou ukázány na obrázku 8.3. Podle očekávání byl největší výpočetní čas vyžadován v případě algoritmu ZIS-E, jelikož ohodnocení strategie probíhá řešením soustavy lineárních rovnic neiteračně. Nejmenší celkový výpočetní čas na vykonání kroku ohodnocení strategie naopak vyžadoval algoritmus ZIS-6. Z výsledků týkajících se výpočetní náročnosti algoritmů je možné usuzovat, že vhodnou volbou počtu iterací k nalezení řešení kroku ohodnocení strategie je možné najít nejméně časově náročné řešení. V této formulaci úlohy celkově nejmenší výpočetní čas k nalezení vhodné aproximace Bellmanovy funkce dosáhl algoritmus ZIS-12, který dokonvergoval v 13-ti iteracích zobecněné iterace strategie. Typický průběh počtu iterací pro řešení kroku ohodnocení strategie generovaný automaticky v algoritmu ZIS-A je prezentován na obrázku 8.4. Z obrázku je patrné, že kromě první iterace $i = 0$ bylo stanoveno $\bar{j} = 53$, neboli 53 postupných iterací. Fixní počet 12-ti postupných iterací je zvolen v algoritmu ZIS-12. Nicméně v obecném případě může být hledání nejlepší volby počtu postupných iterací pro řešení kroku ohodnocení strategie výpočetně náročné. Zároveň ZIS-V vyžaduje definování pevného předpisu pro výpočet počtu iterací pro řešení kroku ohodnocení strategie. Na základě těchto důvodů může navržený algoritmus ZIS-A představovat nejpraktičtější volbu. I přes pozitivní výsledky stojí za připomenutí, že výpočetní nároky pro řídké struktury záleží na implementaci algoritmů.

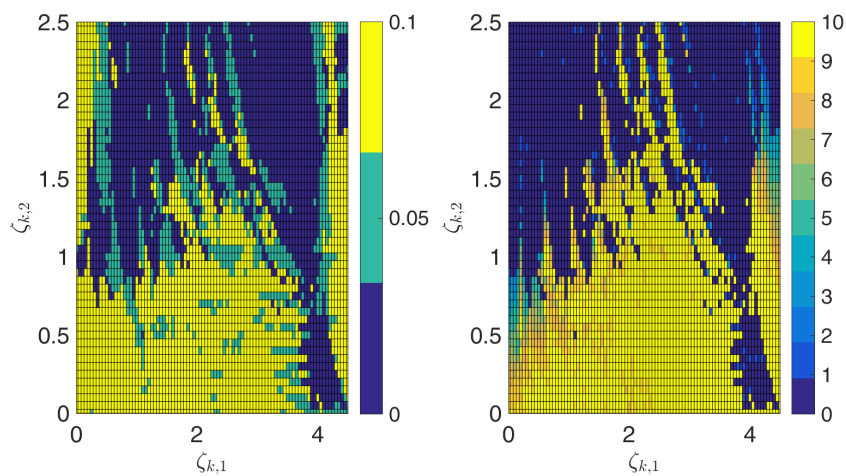


Obrázek 8.4: Typický průběh počtu iterací pro řešení kroku ohodnocení strategie na základě algoritmu ZIS-A pro systém chemostatu.

Aproximace Bellmanovy funkce $\bar{V}_{\bar{p}^{(14)}}$ pro pevnou třetí složku hyper stavu $\zeta_{k,3} = 0.04$ nalezenou pomocí algoritmu ZIS-A ve 14-ti iteracích zobecněné iterace strategie je zobrazena na obrázku 8.5. Funkční hodnoty po částech spojitě funkce jsou barevně odděleny. Nejvyšší hodnoty dosahuje Belmanova funkce pro malé hodnoty $\zeta_{k,1}$, zřejmě z důvodu velkého rozdílu od referenční hodnoty r_1 . Strategie řízení $\bar{\gamma}^{r(14)}$ generující vstupní signál pro pevnou třetí složku hyper stavu $\zeta_{k,3} = 0.04$ nalezená pomocí ZIS-A algoritmu ve 14-ti iteracích zobecněné iterace strategie je zobrazena na obrázku 8.6. Z obrázku je patrný zákon generující příslušné řízení v závislosti na hodnotách hyper stavu ζ_k .

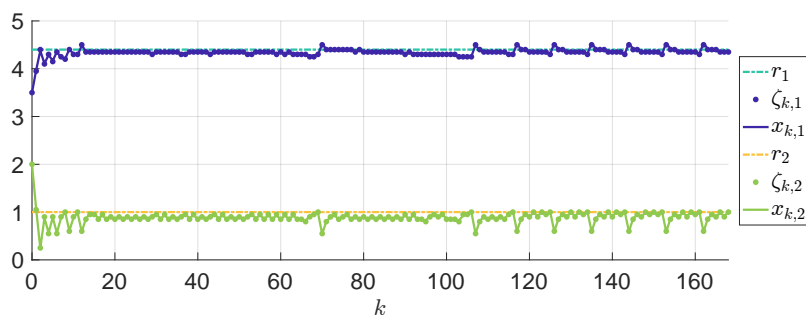


Obrázek 8.5: Bellmanova funkce $\bar{V}_{\bar{p}^r(14)}([\zeta_{k,1}, \zeta_{k,2}, 0.04])$ nalezená pomocí ZIS-A algoritmu pro systém chemostatu.

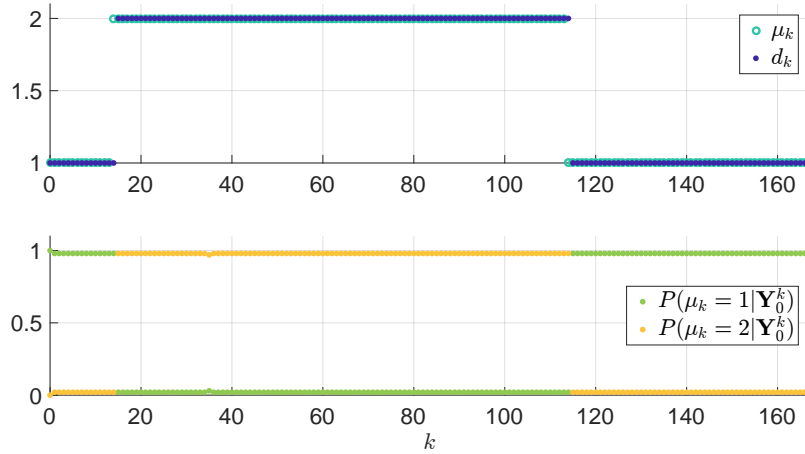


Obrázek 8.6: Strategie $\bar{\gamma}^r(14)([\zeta_{k,1}, \zeta_{k,2}, 0.04])$ generující vstupní signál nalezená pomocí ZIS-A algoritmu pro systém chemostatu.

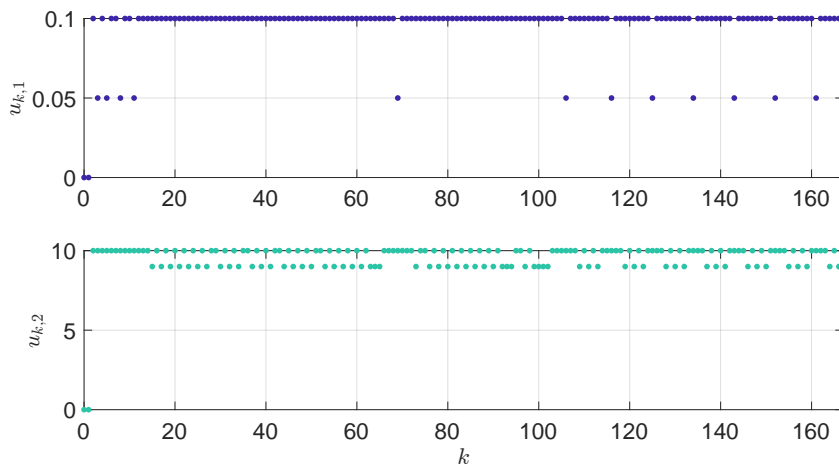
Pro lepší představu vlastností aktivního detektoru chyb a regulátoru navrženého pomocí algoritmu ZIS-A je simulováno chování systému na konečném horizontu jednoho týdne, což představuje 168 časových okamžiků. Počáteční chování systému je bezchybné, neboli $\mu_0 = 1$, a počáteční spojitá část stavu je $\mathbf{x}_0 = [3.5, 2]^T$. Typický průběh trajektorií spojitě části stavu \mathbf{x}_k a prvních dvou složek hyper stavu ζ_k ve srovnání s referenčním signálem \mathbf{r}_k je ukázán na obrázku 8.7. Navzdory kolísání, které je způsobené zřejmě kvantizací spojitého stavového prostoru lze říci, že spojitá část stavu \mathbf{x}_k sleduje referenční hodnotu \mathbf{r}_k . Obrázek 8.8 představuje typické trajektorie diskrétní části stavu μ_k , rozhodnutí d_k a podmíněných pravděpodobností jednotlivých modelů $P(\mu_k | \mathbf{Y}_0^k)$. Z průběhu trajektorií diskrétní části stavu a rozhodnutí je zřejmé, že rozhodnutí je generované správně se zpožděním jednoho kroku. Rozhodnutí je generováno na základě pravděpodobností modelů, které jsou v tomto případě vždy jednoznačně odděleny. Obrázek 8.9 poté ukazuje typické trajektorie obou složek vstupu \mathbf{u}_k , které se ve většině časových okamžiků pohybují v maximálních možných hodnotách. Na základě simulačních výsledků je zřejmé, že navržený aktivní detektor chyb a regulátor splnil cíle řízení a detekce chyb.



Obrázek 8.7: Typický průběh trajektorií referenčního signálu \mathbf{r} , spojitě části stavu \mathbf{x}_k a prvních dvou složek hyper stavu ζ_k pro systém chemostatu.



Obrázek 8.8: Typický průběh trajektorií skutečného diskrétního stavu μ_k , rozhodnutí d_k a pravděpodobností modelů $P(\mu_k = i | \mathbf{Y}_0^k)$ pro $i \in \{1, 2\}$ pro systém chemostatu.



Obrázek 8.9: Typický průběh trajektorií jednotlivých složek vstupního signálu \mathbf{u}_k pro systém chemostatu.

8.1.2 Kyvadlo

Následující příklad se věnuje úloze aktivní detekce chyb pro modely s nepřímo měřitelnou spojitou částí stavu. Návrh řešení je založen na přeformulování úlohy na úlohu s úplnou informací o stavu podle podkapitoly 5.1.2. Informační statistiky jsou získány na základě algoritmu 1, přičemž je uvažována historie ZPB algoritmu $h = 1$. Cílem tohoto příkladu je porovnat tři aktivní detektory chyb navržené různými způsoby řešení optimalizační úlohy s úplnou informací o stavu. Porovnán je aktivní detektor chyb navržený na základě algoritmu 6 učení TD (AFDR-TD), algoritmu 7 QTD využívající ϵ hladovou strategii (AFDR-QTD) a algoritmu 8 využívající neparametrickou aproximaci získanou iterací užitkové funkce (AFDR-NP). Aktivní detektory jsou navrženy off-line, ale mohou být použity on-line.

System kyvadla je ilustrován na obrázku 8.10 a jeho chování může být popsáno pomocí nelineárního spojitého stavového modelu [140]. Takový model může být diskretizován pomocí dopředné Eulerovy metody s periodou vzorkování $T_s = 5 \cdot 10^{-2}$ [s] a linearizován tak, že chování kyvadla v každém časovém okamžiku $k = 0, 1, 2, \dots$ může být popsáno pomocí následujícího lineárního diskrétního stavového modelu

$$\begin{bmatrix} x_{k+1,1} \\ x_{k+1,2} \end{bmatrix} = \begin{bmatrix} 0 & T_s \\ -\frac{T_s g}{l} & 1 - \frac{T_s \beta_{\mu_k}}{\varpi l^2} \end{bmatrix} \begin{bmatrix} x_{k,1} \\ x_{k,2} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{T_s}{\varpi l^2} \end{bmatrix} u_k + \mathbf{G} \mathbf{w}_k, \quad (8.5)$$

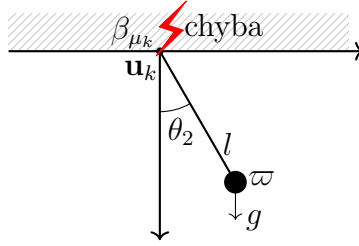
$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}_k + \mathbf{H} \mathbf{v}_k, \quad (8.6)$$

kde $\mathbf{x}_k = [x_{k,1}, x_{k,2}]^T \in \mathbb{R}^2$ je spojitá část stavu systému, $x_{k,1}$ [rad] je úhel natočení kyvadla z dolní rovnovážné polohy, $x_{k,2}$ [rad \cdot s $^{-1}$] je příslušná úhlová rychlost, $\mu_k \in \mathcal{M}$ je index modelu, $u_k \in \mathcal{U}$ [N \cdot m] je vstupní točivý moment aplikovaný v kloubu kyvadla, $y_k \in \mathbb{R}$ je měřený výstup, $\mathbf{w}_k \sim \mathcal{N}([0, 0]^T, \mathbf{I}_2)$ je stavový šum, $\mathbf{v}_k \sim \mathcal{N}(0, 1)$ je šum měření, $g = 9.81$ [m \cdot s $^{-2}$] je tíhové zrychlení, $\varpi = 2$ [kg] je hmotnost kyvadla, $l = 1$ [m] je délka kyvadla a $\beta_1 = 6$ [kg \cdot m 2 \cdot s $^{-1}$] je koeficient tření v bezchybném stavu. Zároveň se předpokládá se množina všech možných vstupů $\mathcal{U} = \{-10, 0, 10\}$, matice stavového šumu $\mathbf{G} = 8 \cdot 10^{-4} \mathbf{I}_2$ a matice šumu měření $\mathbf{H} = 1 \cdot 10^{-3}$. Chování systému se může změnit v důsledku nežádoucí změny tření kloubu kyvadla. Chybné chování může být modelováno změnou koeficientu tření na hodnotu $\beta_2 = 6.2$ [kg \cdot m 2 \cdot s $^{-1}$]. Množina všech možných indexů modelů je tudíž definována jako $\mathcal{M} = \{1, 2\}$. Trajektorie indexu modelu μ_k se vyvíjí podle matice pravděpodobnosti přechodů

$$\mathbf{P} = \begin{bmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{bmatrix}. \quad (8.7)$$

V počátečním časovém okamžiku je chování systému bezchybné, neboli $\mu_0 = 1$ a $P(\mu_0 = 1) = 1$. Počáteční spojitá část stavu může být popsána normálním rozdělením pravděpodobnosti $\mathbf{x}_0 \sim \mathcal{N}([0, 0]^T, 2 \cdot 10^{-4} \mathbf{I}_2)$.

Cílem úlohy je najít takový aktivní detektor chyb (4.5) definovaný strategií ρ generující rozhodnutí $d_k \in \mathcal{M}$ a vstup \mathbf{u}_k tak, že je minimalizováno kritérium kvality (4.7) s diskont-



Obrázek 8.10: Ilustrační schéma kyvadla.

ním faktorem $\eta = 0.9$ a s následující ztrátovou funkcí penalizující nesprávná rozhodnutí

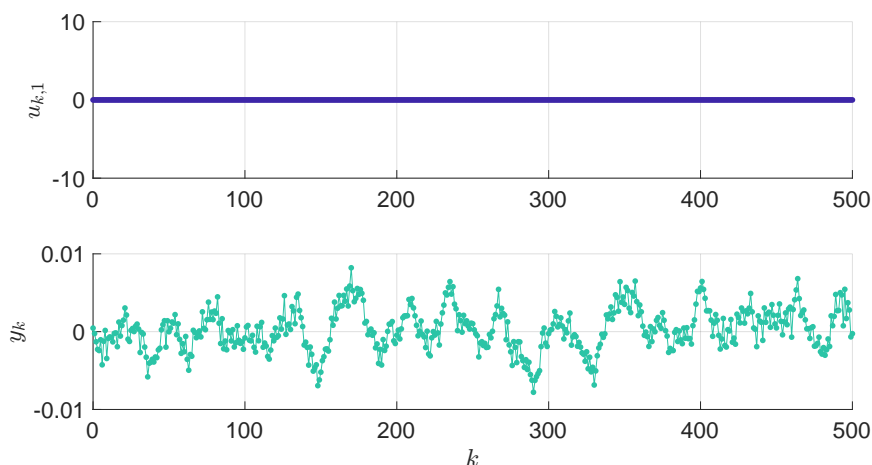
$$L^d(\mu_k, d_k) = \begin{cases} 0 & \text{pokud } d_k = \mu_k, \\ 1 & \text{jinak.} \end{cases} \quad (8.8)$$

V případě $d_k = \mu_k$ je penalizace nulová, avšak v opačném případě chybné detekce $d_k \neq \mu_k$ je takové rozhodnutí penalizováno hodnotou jedna.

Informační statistiky jsou obsaženy v hyperstavu ξ_k , který je definovaný v (5.53). Lineární aproximaci Bellmanovy funkce (6.42) pro AFDR-TD využívá $n_\omega = 24$ normalizovaných gaussovských bázových funkcí (6.43) s návrhovým parametrem $\delta_j^\xi = 0.5$ pro všechna $j = 1, 2, \dots, n_\omega$ a středy ξ_j^c , které byly heuristicky vybrány z konečné diskrétní množiny hyperstavů $\mathcal{G}^s \subset \mathcal{G}$ získané na základě simulace systému pod vhodně zvolenými vstupními signály. Pro AFDR-TD a AFDR-QTD byl shodně zvolen parametr délky kroku $\kappa_k^t = \kappa_k^q = \frac{1000}{2000 + \kappa_k^q}$, $\kappa_0^q = 0$ a parametr TD $\lambda = 0.4$. V případě lineární aproximace Q-funkce (6.57) pro AFDR-QTD je zvoleno $n_\theta = 72$ normalizovaných gaussovských bázových funkcí, které reflektují množinu všech přípustných vstupů \mathcal{U} o mohutnosti tři. Pravděpodobnost generování náhodného vstupu ϵ -hladové strategie je zvolena jako $\epsilon = 0.03$. Ukončovací podmínkou pro TD a QTD je dosažení maximálního počtu iterací $N_{\text{TD}}^{\text{max}} = N_{\text{QTD}}^{\text{max}} = 10000$. Počáteční váhy $\omega^{(0)}$ a $\theta^{(0)}$ jsou nulové. Neparametrická architektura pro AFDR-NP odpovídá (6.65) se vzorky hyperstavu odpovídající předchozím vygenerovaným vzorkům $\xi_j^s \in \mathcal{G}^s$, kde $j = 1, 2, \dots, N_\xi^s$, $N_\xi^s = 2505$. Jádrová funkce odpovídá (6.66) s návrhovým parametrem $\delta^K = 0.5$. Přibližné hodnoty Bellmanovy funkce V_j^s pro diskrétní hyperstavy ξ_j^s jsou vypočítány iterací užitkové funkce s ukončovací podmínkou (6.68) a mezí přesnosti $\epsilon_{\text{VI}} = 1 \cdot 10^{-2}$. V případě, že ukončovací podmínka (6.68) není splněna ani do maximálního počtu $N_{\text{VI}}^{\text{max}} = 50$ iterací, iterace užitkové funkce je ukončena. Střední hodnota v (6.47) pro AFDR-TD a v (6.70) pro AFDR-NP je aproximována pomocí $N_{\text{MC}} = 100$ MC simulací.

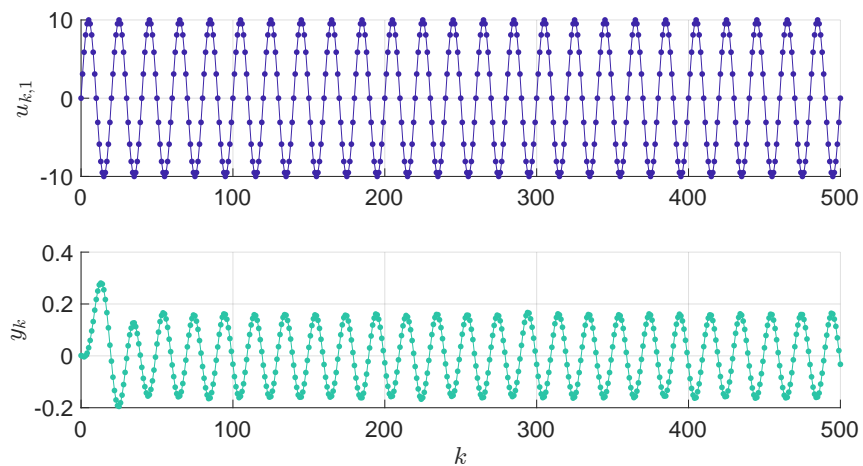
Chování navržených aktivních detektorů chyb AFDR-TD, AFDR-QTD a AFDR-NP bylo porovnáno s chováním pasivního detektoru chyb (PFDR) a chováním aktivního detektoru chyb využívající generátor vstupního sinusového signálu s amplitudou 10 a frekvencí 1 [Hz] (AFDR-SIN). Simulace chování systému byla provedena na konečném časovém horizontu $k = 0, 1, \dots, 500$. Typický průběh trajektorií vstupu a výstupu systému je pro PFDR zobrazen na obrázku 8.11, pro AFDR-SIN na obrázku 8.12, pro AFDR-TD na obrázku 8.13, pro AFDR-QTD na obrázku 8.14 a pro AFDR-NP na obrázku 8.15. V případě

nulového vstupu PFDR se hodnoty výstupu systému pohybují kolem nulové hodnoty v důsledku nejistot v systému. Pro sinusový signál generovaný AFDR-SIN je tvar výstupního signálu opět sinusový v důsledku linearitý systému. Aktivní detektor chyb navržený pomocí učení QTD má téměř pravidelný průběh budicího vstupního signálu, který v intervalech přeskakuje mezi minimální hodnotou vstupu $\min\{u_k\} = 10$ a maximální hodnotou vstupu $\max\{u_k\} = 10$ s periodou 40 kroků, což odpovídá časovému úseku 2 [s]. Tvary vstupních signálů pro AFDR-TD a AFDR-NP jsou ve srovnání s tvarem pro AFDR-QTD více nepravidelné a hodnoty vstupu rovněž nabývají hodnoty $u_k = 0$.

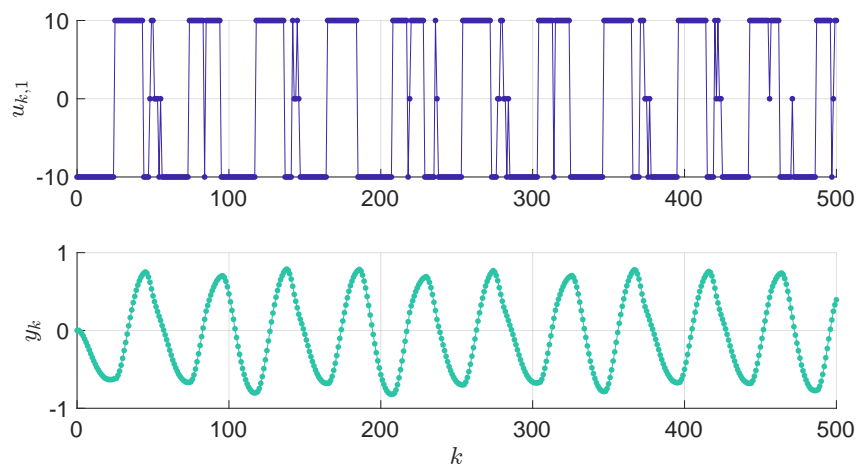


Obrázek 8.11: Typický průběh trajektorií vstupu u_k a výstupu y_k pro PFDR pro systém kyvadla.

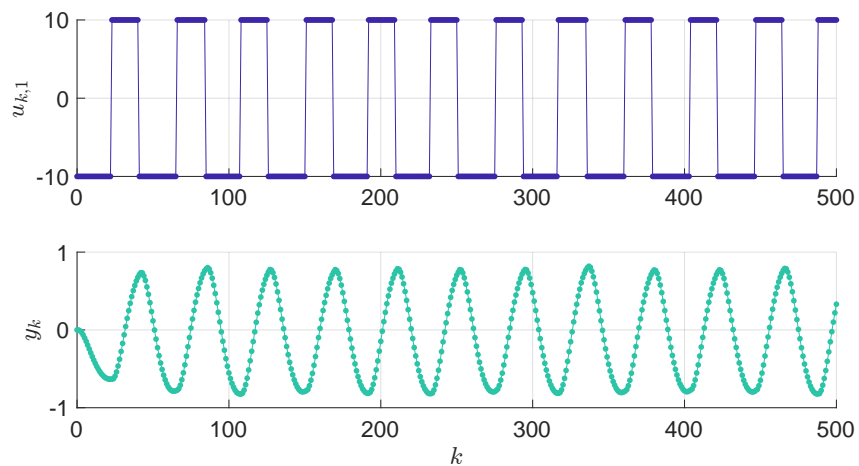
K ohodnocení kvality navržených detektorů chyb poslouží porovnání typických trajektorií skutečného indexu modelu, rozhodnutí a podmíněných pravděpodobností indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$, které jsou zobrazeny pro PFDR na obrázku 8.16, pro AFDR-SIN na obrázku 8.17, pro AFDR-TD na obrázku 8.18, pro AFDR-QTD na obrázku 8.19 a pro AFDR-NP na obrázku 8.20. Ze srovnání těchto trajektorií je patrné, že v případě pasivní detekce chyb dochází na rozdíl od aktivní detekce chyb k velkému množství chybných rozhodnutí, neboli $d_k \neq \mu_k$. Generátor rozhodnutí PFDR generuje chybná rozhodnutí, jelikož po uplynutí přibližně 110 časových okamžiků se podmíněné pravděpodobnosti dvou možných indexů modelů $P(\mu_k = 1 | \mathbf{Z}_0^k)$ a $P(\mu_k = 2 | \mathbf{Z}_0^k)$ ustálí na hodnotě pravděpodobnosti 0.5, a proto je těžké rozlišit mezi možnými chováními systému. Naproti tomu budicí vstupní signál aktivních detektorů chyb zajistí, že podmíněné pravděpodobnosti indexů modelů $P(\mu_k = 1 | \mathbf{Z}_0^k)$ a $P(\mu_k = 2 | \mathbf{Z}_0^k)$ jsou odděleny. V případě AFDR-SIN však nejsou hodnoty $P(\mu_k = 1 | \mathbf{Z}_0^k)$ a $P(\mu_k = 2 | \mathbf{Z}_0^k)$ odděleny do takové míry, jako v případě vhodně navržených detektorů AFDR-TD, AFDR-QTD a AFDR-NP. Nicméně trajektorie rozhodnutí pro všechny aktivní detektory chyb jsou na první pohled velmi podobné, a tak je zapotřebí srovnat výkonnost na základě většího počtu simulací. Pro toto porovnání je chování systému na konečném časovém horizontu $k = 0, 1, \dots, 500$ ohodnoceno pomocí $N_{MC}^{\text{eval}} = 5000$



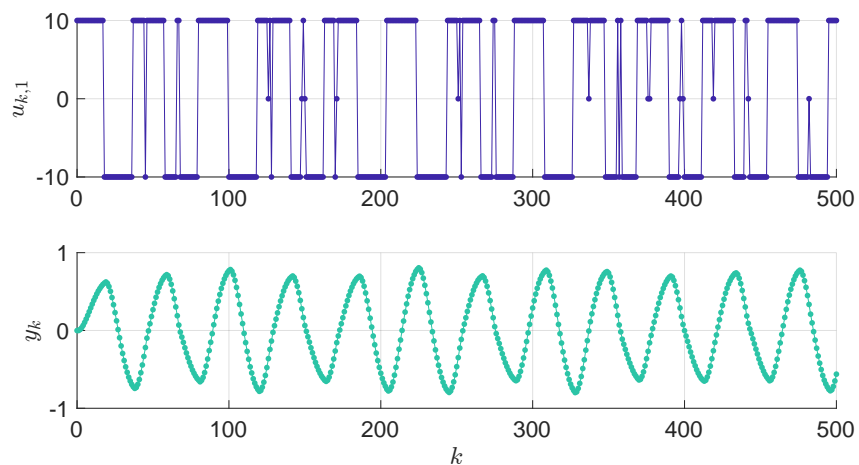
Obrázek 8.12: Typický průběh trajektorií vstupu u_k a výstupu y_k pro AFDR-SIN pro systém kyvadla.



Obrázek 8.13: Typický průběh trajektorií vstupu u_k a výstupu y_k pro AFDR-TD pro systém kyvadla.



Obrázek 8.14: Typický průběh trajektorií vstupu u_k a výstupu y_k pro AFDR-QTD pro systém kyvadla.



Obrázek 8.15: Typický průběh trajektorií vstupu u_k a výstupu y_k pro AFDR-NP pro systém kyvadla.

Tabulka 8.1: Odhad hodnoty kritéria \hat{J} , výběrového rozptylu $\text{var } \hat{J}$ a průměrného výpočetního času jedné simulace T^{sim} různých detektorů chyb na základě $N_{\text{MC}}^{\text{eval}} = 5000$ MC simulací pro systém kyvadla.

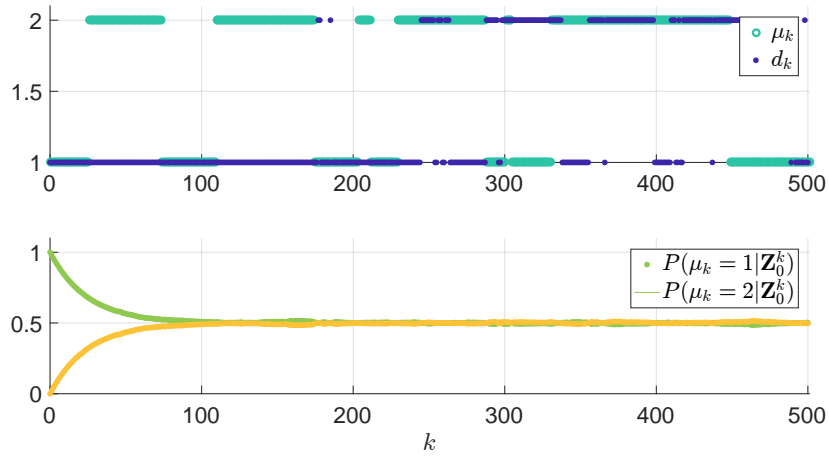
Generátor vstupního signálu	\hat{J}	$\text{var } \hat{J}$	T^{sim} [s]
PFDR	1.3676	$9 \cdot 10^{-4}$	0.1582
AFDR-SIN	1.1399	$6 \cdot 10^{-4}$	0.1576
AFDR-TD	1.0806	$5 \cdot 10^{-4}$	34.6147
AFDR-QTD	1.0922	$6 \cdot 10^{-4}$	0.2430
AFDR-NP	1.0452	$5 \cdot 10^{-4}$	66.9419

MC simulací. Odhad hodnoty kritéria

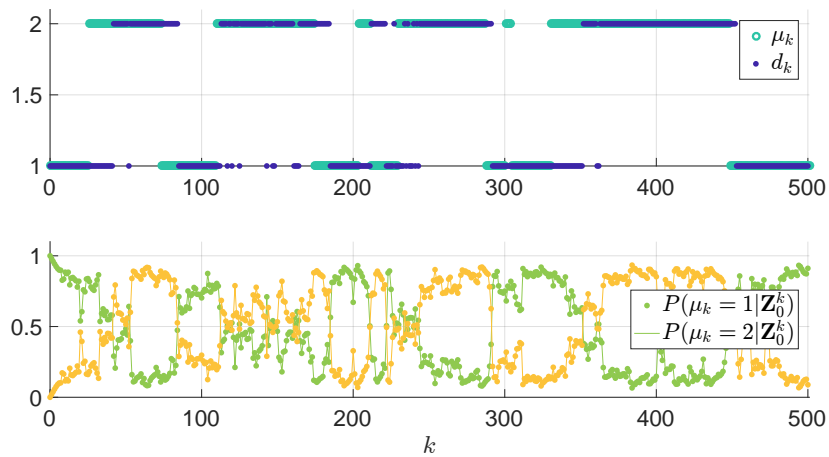
$$\hat{J} = \frac{1}{N_{\text{MC}}^{\text{eval}}} \sum_{i=1}^{N_{\text{MC}}^{\text{eval}}} J_i \quad (8.9)$$

a jeho výběrový rozptyl $\text{var } \hat{J}$ získaný na základě metody bootstrap [141] jsou uvedeny v tabulce 8.1. Nejmenší hodnotu odhadu kritéria \hat{J} a tedy i nejlepší detekce chyb dosáhl AFDR-NP, a naopak pasivní detektor chyb PFDR vykázal nejvyšší hodnotu \hat{J} . Z tabulky 8.1 je zároveň patrné, že všechny navržené aktivní detektory chyb AFDR-TD, AFDR-QTD a AFDR-NP mají menší hodnotu kritéria než heuristicky zvolený aktivní detektor chyb AFDR-SIN. Hodnoty výběrového rozptylu jsou pro všechny detektory chyb srovnatelné. Tabulka 8.1 zároveň obsahuje průměrný výpočetní čas jedné simulace T^{sim} na konečném časovém horizontu $k = 0, 1, \dots, 500$ pro jednotlivé detektory chyb. Ze získaných hodnot je patrné, že nejnižších simulačních časů logicky dosahuje PFDR a AFDR-SIN, jelikož je vstup u_k v každém časovém okamžiku k generován přímo na základě známé heuristiky. V případě PFDR je vstupní signál nulový a v případě AFDR-SIN se využívá jednoduchého funkčního předpisu goniometrické funkce sinus. Podobného výpočetního času ovšem dosahuje také AFDR-QTD, jelikož vnitřní mechanismus navržené strategie neobsahuje výpočet střední hodnoty Bellmanovy funkce, na rozdíl od AFDR-TD a AFDR-NP. Pro upřesnění se výpočet střední hodnoty Bellmanovy funkce vyžaduje v generátoru vstupního signálu (6.47) pro AFDR-TD a (6.70) pro AFDR-NP. Přestože hodnota \hat{J} pro AFDR-QTD je o něco vyšší než stejná hodnota pro AFDR-TD, v některých aplikacích může být stěžejním bodem návrhu výpočetní čas, a proto by bylo vhodné použít právě AFDR-QTD.

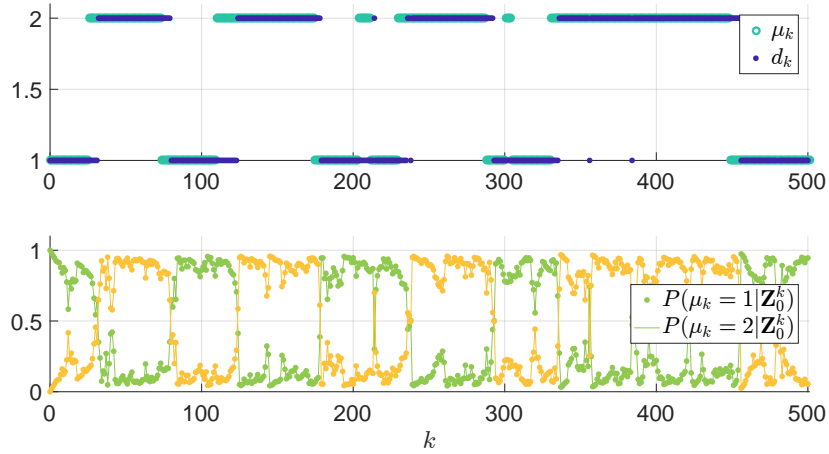
Dosažené výsledky týkající se srovnání pasivní a aktivní detekce chyb v případě systému kyvadla jsou logické. Zvýšené tření v kloubu kyvadla může být pouze stěží odhaleno, pokud zůstává kyvadlo nehybně v jedné poloze. Jestliže se kyvadlo uvede do pohybu, změnu tření v kloubu kyvadla je možné důsledkem snížené hybnosti kyvadla rozpoznat. Proto je vhodné v takové úloze použít aktivní detektor chyb.



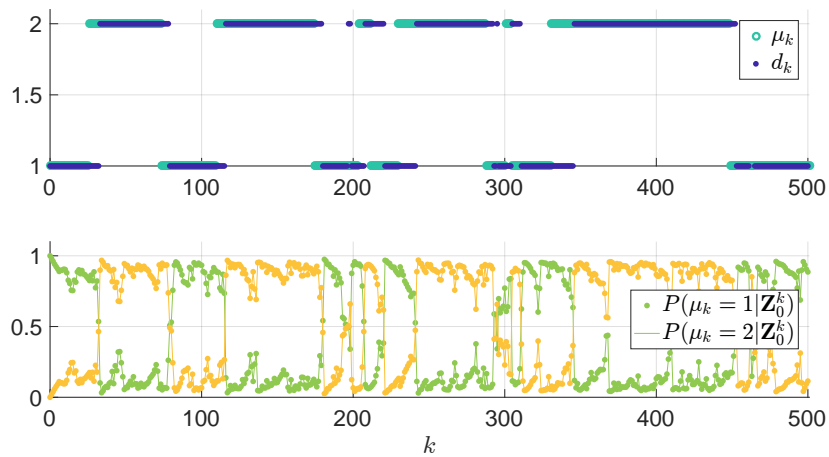
Obrázek 8.16: Typický průběh trajektorií indexu modelu μ_k , rozhodnutí d_k a podmíněných pravděpodobností indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$ pro PFDR pro systém kyvadla.



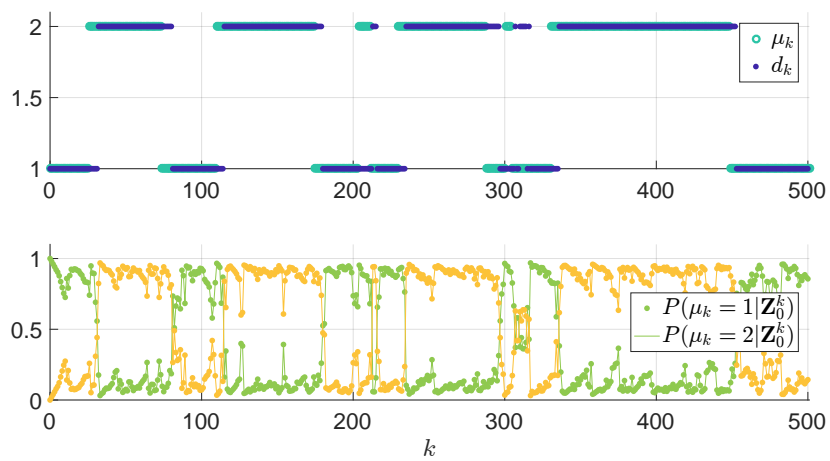
Obrázek 8.17: Typický průběh trajektorií indexu modelu μ_k , rozhodnutí d_k a podmíněných pravděpodobností indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$ pro AFDR-SIN pro systém kyvadla.



Obrázek 8.18: Typický průběh trajektorií indexu modelu μ_k , rozhodnutí d_k a podmíněných pravděpodobností indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$ pro AFDR-TD pro systém kyvadla.



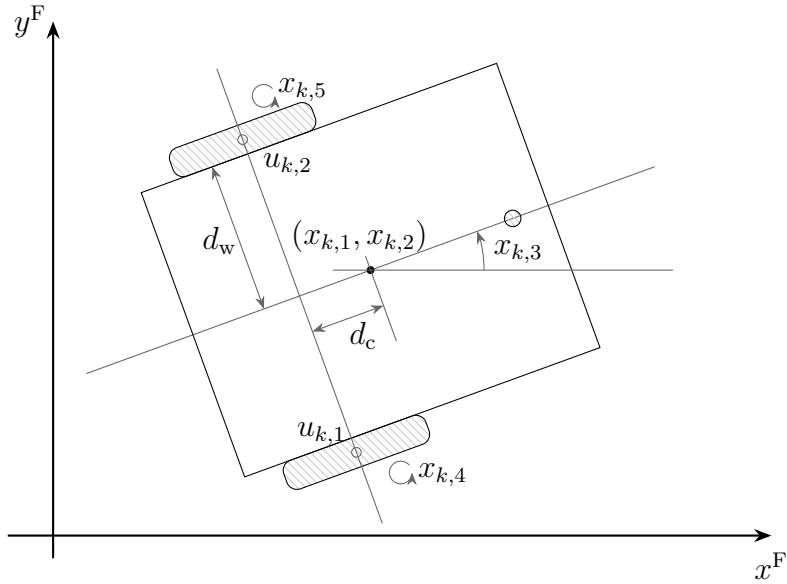
Obrázek 8.19: Typický průběh trajektorií indexu modelu μ_k , rozhodnutí d_k a podmíněných pravděpodobností indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$ pro AFDR-QTD pro systém kyvadla.



Obrázek 8.20: Typický průběh trajektorií indexu modelu μ_k , rozhodnutí d_k a podmíněných pravděpodobností indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$ pro AFDR-NP pro systém kyvadla.

8.1.3 Mobilní robot

V tomto numerickém příkladě je uvažován mobilní robot s diferenciálním podvozkem [142]. Na rozdíl od obvyklé úlohy sledování referenční trajektorie je cílem detekovat náhlé změny v tření. Systém je popsán vícemodelovým přístupem s nepřímo měřitelnou spojitou částí stavu. Proto je zapotřebí optimalizační úlohu nejprve přeformulovat na úlohu s úplnou informací o stavu pomocí metody odhadu stavu. Následně je možné najít aktivní detektor chyb pomocí zpětnovazebního učení. Cílem tohoto příkladu je porovnat způsoby přeformulování využívající dva rozdílné algoritmy odhadu stavu, konkrétně algoritmus 2 (AFDR-EKF) využívající EKF a algoritmus 3 (AFDR-PF) založený na bootstrap PF. Aktivní detektor chyb je pro obě dvě porovnávané varianty navržen na základě algoritmu 6 zpětnovazebního učení pomocí TD.



Obrázek 8.21: Ilustrační schéma mobilního robotu.

Ilustrační schéma mobilního robotu je uvedeno na obrázku 8.21. Systém mobilního robotu se skládá z platformy tvořící tělo robotu, dvou nezávislých hnaných zadních kol a volného předního kola trnu. Robot se pohybuje v rovině a změna pohybu a orientace je dosažena použitím vhodného kroutícího momentu zadních hnacích kol. Necht je uvažována následující spojitá část stavu systému $\mathbf{x}_k = [x_{k,1}, x_{k,2}, x_{k,3}, x_{k,4}, x_{k,5}]^T$, kde $(x_{k,1}, x_{k,2})$ [m] jsou kartézské souřadnice těžiště robotu v inerciální vztahné soustavě, $x_{k,3}$ [rad] je úhel natočení, který svírá podélná osa robotu s osou x^F vztahné soustavy, $x_{k,4}$ [rad · s⁻¹] je úhlová rychlost pravého zadního hnacího kola a $x_{k,5}$ [rad · s⁻¹] je úhlová rychlost levého zadního hnacího kola. Uvažují se dvě možná chování systému jednoznačně definovaná indexem modelu $\mu_k \in \mathcal{M} = \{1, 2\}$, přičemž bezchybné chování odpovídá $\mu_k = 1$. Chybné chování představuje změnu tření v ložiskách robotu a odpovídá $\mu_k = 2$. Trajektorie stavu může být

popsána následujícím nelineárním diskretním stochastickým stavovým modelem

$$\mathbf{x}_{k+1} = \mathbf{x}_k + T_s \begin{bmatrix} \mathbf{S}_k \\ -\mathbf{M}^{-1} (\mathbf{V}_k + \mathbf{S}_k^T \mathbf{\Lambda}_{\mu_k} \mathbf{S}_k) \end{bmatrix} \begin{bmatrix} x_{k,4} \\ x_{k,5} \end{bmatrix} + T_s \begin{bmatrix} \mathbf{0}_{3 \times 2} \\ \mathbf{M}^{-1} \end{bmatrix} \mathbf{u}_k + \mathbf{G} \mathbf{w}_k, \quad (8.10)$$

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{H} \mathbf{v}_k, \quad (8.11)$$

kde $\mathbf{0}_{3 \times 2}$ je 3×2 matice nul, $\mathbf{\Lambda}_{\mu_k}$ je 3×3 diagonální matice koeficientů tření a \mathbf{S}_k , \mathbf{V}_k spolu s \mathbf{M} jsou definovány jako

$$\mathbf{S}_k = \begin{bmatrix} \frac{r}{2} \cos(x_{k,3}) & \frac{r}{2} \cos(x_{k,3}) \\ \frac{r}{2} \sin(x_{k,3}) & \frac{r}{2} \sin(x_{k,3}) \\ \frac{r}{2d_w} & -\frac{r}{2d_w} \end{bmatrix}, \mathbf{V}_k = \begin{bmatrix} 0 & \frac{m_p r^3 d_c (x_{k,4} - x_{k,5})}{4d_w^2} \\ \frac{m_p r^3 d_c (x_{k,4} - x_{k,5})}{4d_w^2} & 0 \end{bmatrix}, \quad (8.12)$$

$$\mathbf{M} = \begin{bmatrix} \frac{r}{4d_w^2} (m_r d_w^2 + I_r) + I_w & \frac{r}{4d_w^2} (m_r d_w^2 - I_r) \\ \frac{r}{4d_w^2} (m_r d_w^2 - I_r) & \frac{r}{4d_w^2} (m_r d_w^2 + I_r) + I_w \end{bmatrix}, \quad (8.13)$$

kde $\mathbf{u}_k = [u_{k,1}, u_{k,2}]^T$ [$\text{kg} \cdot \text{m}^2 \cdot \text{s}^{-2}$] je vstup systému, $u_{k,1}$ je točivý moment pravého zadního kola, $u_{k,2}$ je točivý moment levého zadního kola, $T_s = 5 \cdot 10^{-2}$ [s] je perioda vzorkování, $d_c = 0.2$ [m] je vzdálenost mezi těžištěm a středem spojnice mezi zadními koly, $d_w = 0.5$ [m] je polovina nejmenší vzdálenosti mezi zadními koly, $r = 0.15$ [m] je poloměr kol, $m_p = 30$ [kg] je hmotnost platformy, $m_r = 32$ [kg] je celková hmotnost robotu, $I_r = 17.205$ [$\text{kg} \cdot \text{m}^2$] je moment setrvačnosti robotu, $I_w = 0.005$ [$\text{kg} \cdot \text{m}^2$] je moment setrvačnosti kol, $\mathbf{G} = \text{diag}([10^{-4}, 10^{-4}, 10^{-4}, 10^{-3}, 10^{-3}])$ je matice stavového šumu, $\mathbf{H} = \text{diag}([10^{-2}, 10^{-2}, 10^{-2}, 10^{-3}, 10^{-3}])$ je matice šumu měření a $\mathbf{\Lambda}_1 = \text{diag}([15, 15, 15])$ je matice koeficientů tření v případě bezchybného chování systému. Vstup \mathbf{u}_k systému může nabývat hodnot z množiny \mathcal{U} , která je definovaná jako $\mathcal{U} = \{-1, 0, 1\} \times \{-1, 0, 1\}$. Trajektorie indexu modelu μ_k se vyvíjí podle matice pravděpodobnosti přechodů

$$\mathbf{P} = \begin{bmatrix} 0.96 & 0.04 \\ 0.04 & 0.96 \end{bmatrix}. \quad (8.14)$$

Počáteční chování systému je bezchybné, neboli $P(\mu_0 = 1) = 1$, a počáteční spojitá část stavu může být popsána normálním rozdělením pravděpodobnosti $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}_{5 \times 1}, 1 \cdot 10^{-4} \mathbf{I}_5)$. Chybné chování systému představuje změnu matice koeficientů tření na $\mathbf{\Lambda}_2 = \text{diag}([20, 20, 20])$.

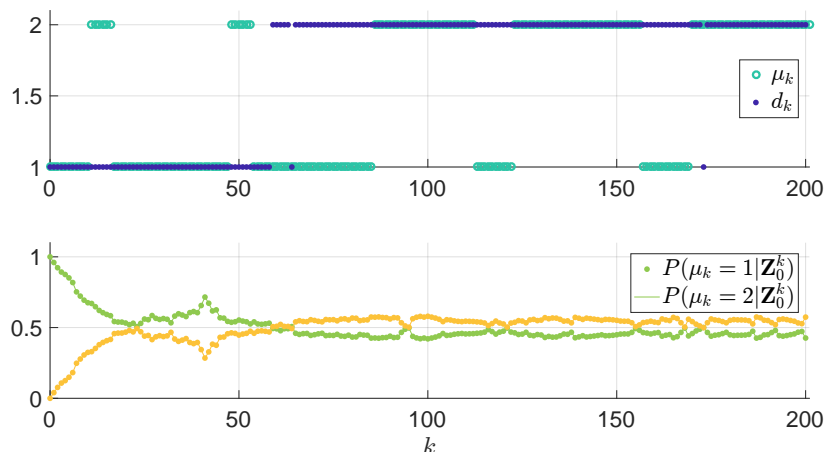
Cílem úlohy je navrhnout takový aktivní detektor chyb (4.5) definovaný strategií ρ generující rozhodnutí $d_k \in \mathcal{M}$ a vstup \mathbf{u}_k , který minimalizuje kritérium kvality (4.7). Návrhové kritérium je definované diskontním faktorem $\eta = 0.9$ a ztrátovou funkcí penalizující nesprávná rozhodnutí

$$L^d(\mu_k, d_k) = \begin{cases} 0 & \text{pokud } d_k = \mu_k, \\ 1 & \text{jinak.} \end{cases} \quad (8.15)$$

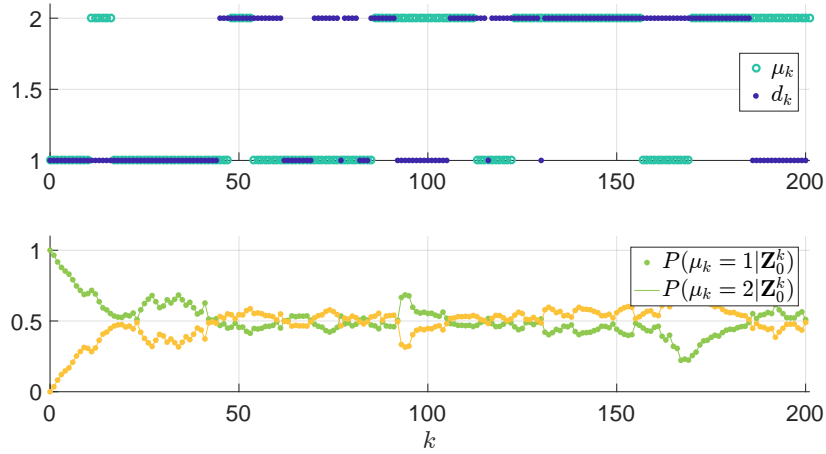
Hyperstav ξ_k definovaný v (5.53) obsahující informační statistiky stavu $\mathbf{s}_k = [\mathbf{x}_k^T, \mu_k]^T$ je získán dvěma způsoby. První způsob je založen na odhadu stavu pomocí EKF a ZPB

s historií $h = 1$. V druhém případě se využívá bootstrap PF s pevným počtem $N_p = 3000$ částic. Bellmanova funkce je pro oba aktivní detektory chyb nahrazena lineární parametrickou aproximací (6.42) využívající $n_\omega = 110$ normalizovaných gaussovských báзовých funkcí (6.43) s návrhovým parametrem $\delta_j^\xi = 0.5$ pro všechna $j = 1, 2, \dots, n_\omega$ a středy ξ_j^c , které byly heuristicky vybrány z konečné diskrétní množiny hyperstavů $\mathcal{G}^g \subset \mathcal{G}$ získané na základě simulace systému s vhodně zvolenými vstupními signály. Parametr délky kroku je $\kappa_k^t = \frac{1000}{2000 + \kappa_{k-1}^t}$, $\kappa_0^t = 0$ a parametr TD je $\lambda = 0.4$. Ukončovací podmínka algoritmu učení TD je dosažení maximálního počtu iterací $N_{TD}^{\max} = 200$. Počáteční váhy $\omega^{(0)}$ jsou nulové. Střední hodnota v (6.47) je aproximována pomocí $N_{MC} = 60$ MC simulací.

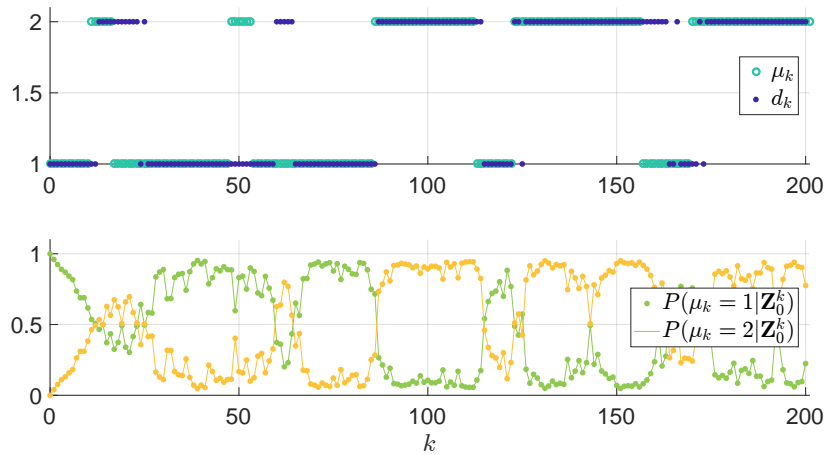
Navržené aktivní detektory chyb AFDR-EKF a AFDR-PF jsou vzájemně porovnány na konečném časovém horizontu 201 kroků. Zároveň jsou navržené aktivní detektory srovnány s pasivním detektorem chyb (PFDR-EKF) využívající algoritmus 2 podobně jako AFDR-EKF a s pasivním detektorem chyb (PFDR-PF) využívající algoritmus 3 podobně jako AFDR-PF. Pasivní detektory chyb používají generátor vstupního signálu v otevřené smyčce, který generuje nulový signál. Typický průběh trajektorií indexu modelu μ_k , rozhodnutí d_k a podmíněných pravděpodobností indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$ je pro PFDR-EKF zobrazen na obrázku 8.22, pro PFDR-PF na obrázku 8.23, pro AFDR-EKF na obrázku 8.24 a pro AFDR-PF na obrázku 8.25. Z průběhů trajektorií je patrné, že v případě pasivní detekce chyb dochází k velkému množství chybných rozhodnutí. Pro PFDR-EKF a PFDR-PF se totiž podmíněné pravděpodobnosti indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$ přiblíží k hodnotě 0.5, kde zůstanou až do konce měření. V případě aktivní detekce chyb dochází k buzení systému pomocí vhodně navrženého vstupu, a tím jsou podmíněné pravděpodobnosti dvou možných indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$ rozděleny, jak je vidět na obrázcích 8.24 a 8.25. Nicméně nezanedbatelná část rozhodnutí je chybná.



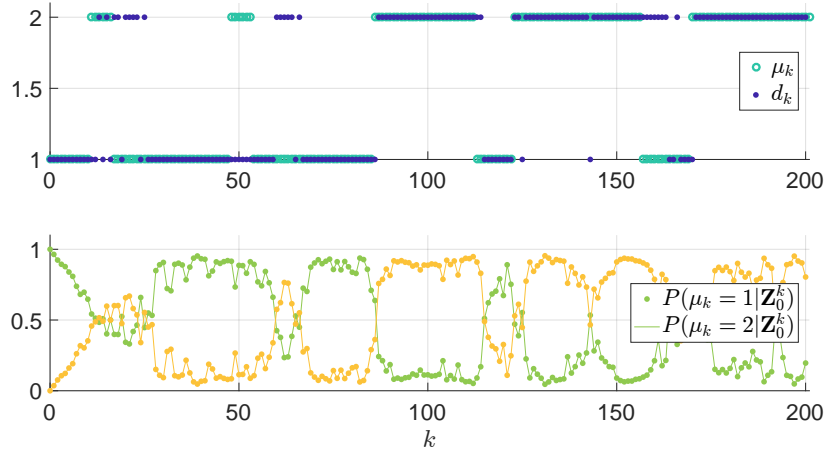
Obrázek 8.22: Typický průběh trajektorií indexu modelu μ_k , rozhodnutí d_k a podmíněných pravděpodobností indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$ pro PFDR-EKF pro systém mobilního robotu.



Obrázek 8.23: Typický průběh trajektorií indexu modelu μ_k , rozhodnutí d_k a podmíněných pravděpodobností indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$ pro PFDR-PF pro systém mobilního robotu.



Obrázek 8.24: Typický průběh trajektorií indexu modelu μ_k , rozhodnutí d_k a podmíněných pravděpodobností indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$ pro AFDR-EKF pro systém mobilního robotu.



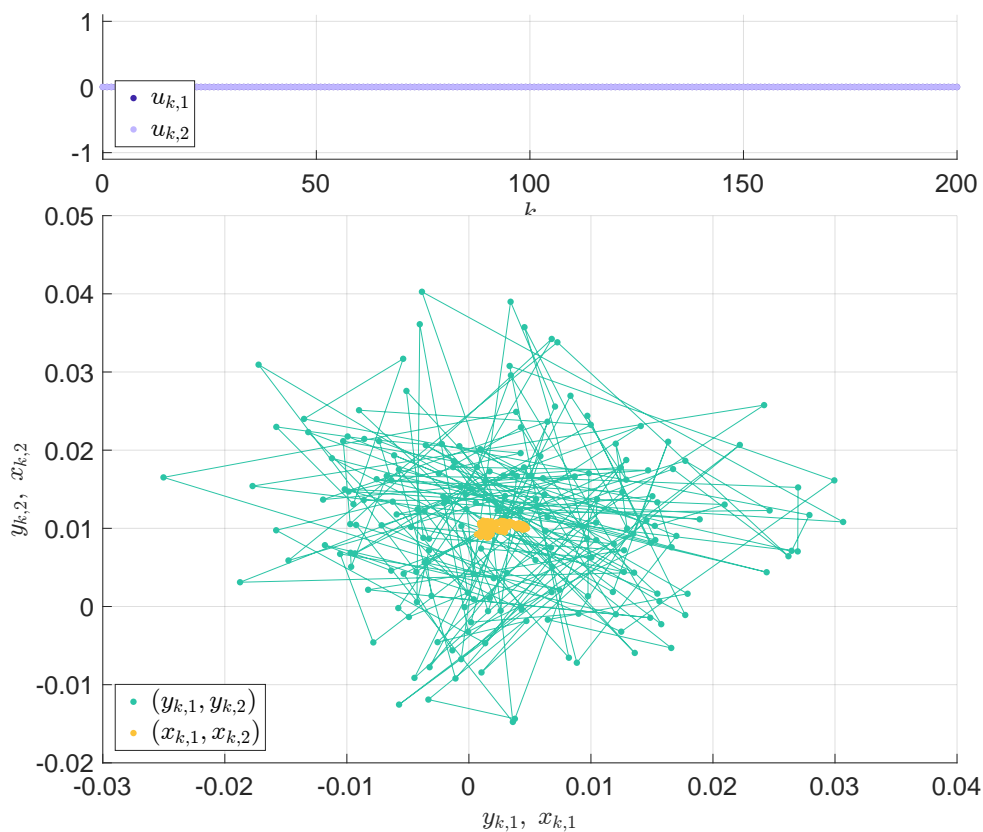
Obrázek 8.25: Typický průběh trajektorií indexu modelu μ_k , rozhodnutí d_k a podmíněných pravděpodobností indexů modelů $P(\mu_k | \mathbf{Z}_0^k)$ pro AFDR-PF pro systém mobilního robotu.

Na obrázcích 8.26, 8.27 a 8.28 je zobrazen typický průběh trajektorií vstupu \mathbf{u}_k a kartézských souřadnic spojité části stavu $(x_{k,1}, x_{k,2})$ a výstupu $(y_{k,1}, y_{k,2})$ mobilního robotu pro pasivní detektory, AFDR-EKF a AFDR-PF. Z uvedených trajektorií spojité části stavu a výstupu na obrázku 8.26 je patrné, že šum měření má výrazný dopad na hodnoty měřeného výstupu. Podle očekávání je v případě pasivní detekce chyba poloha robotu ovlivněna pouze nepatrně náhodným šumem. V případě aktivní detekce chyba způsobuje navržený budicí vstupní signál pohyb robotu ve směru osy x^F , který je zřejmý z obrázků 8.27 a 8.28. Vstupní signál navržený pomocí AFDR-EKF nabývá hodnot z množiny $\{[0, 0], [1, 1], [-1, -1]\}$ a vstupní signál navržený pomocí AFDR-PF nabývá hodnot z množiny $\{[0, 0], [1, 1]\}$.

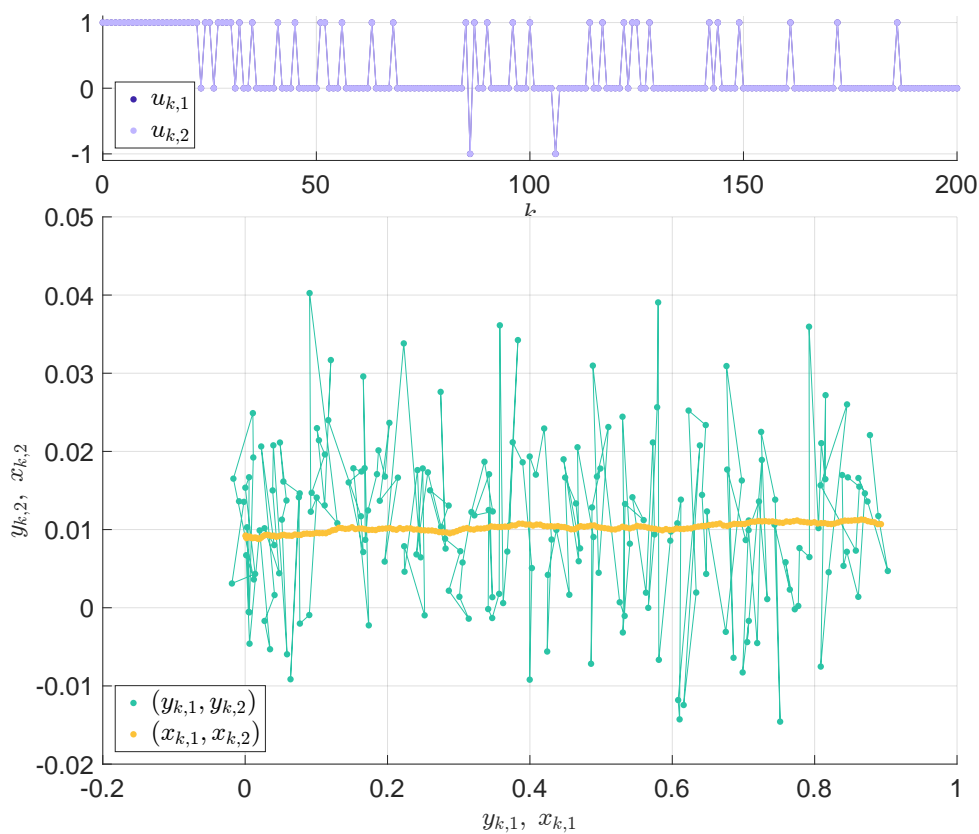
Chování uvedených detektorů chyb bylo dále vzájemně srovnáno pomocí $N_{MC}^{eval} = 1000$ MC simulací na konečném časovém horizontu 201 kroků. Z hlediska odhadu hodnoty kritéria \hat{J} dosáhly aktivní detektory chyb podle očekávání lepšího chování v porovnání s pasivními detektory chyb, jelikož v případě pasivní detekce chyb buzení systému stavovým šumem není dostatečné. Ve srovnání s celkově nejlepším AFDR-PF byl z hlediska kvality detekce chyb AFDR-EKF horší o 1.5 procenta, PFDR-EKF o 15.1 procent a PFDR-PF o 15,6 procent. Kvalitu detekce aktivních detektorů chyb by zcela jistě bylo možné zlepšit vhodnějším výběrem aproximace Bellmanovy funkce, který ovšem není triviální. Dalším faktorem ovlivňujícím kvalitu detekce chyb je samotná volba návrhového kritéria.

Kromě kvality detekce chyb byl sledován výpočetní čas jedné simulace T^{sim} na daném časovém horizontu. PFDR-EKF dosáhl nejnižšího výpočetního času $T_{PFDR-EKF}^{sim} = 0.2558$ [s]. Druhým nejlepším byl PFDR-PF s výpočetním časem $T_{PFDR-PF}^{sim} = 1.1663$ [s]. Výpočetní čas pro AFDR-EKF byl $T_{AFDR-EKF}^{sim} = 48.3697$ [s] a pro AFDR-PF byl $T_{AFDR-PF}^{sim} = 1.0437 \cdot 10^3$ [s]. Z dosažených výsledků lze pozorovat značně vyšší požadavky na výpočetní čas pro oba aktivní detektory chyb. To je způsobeno určováním budicího vstupu na základě zís-

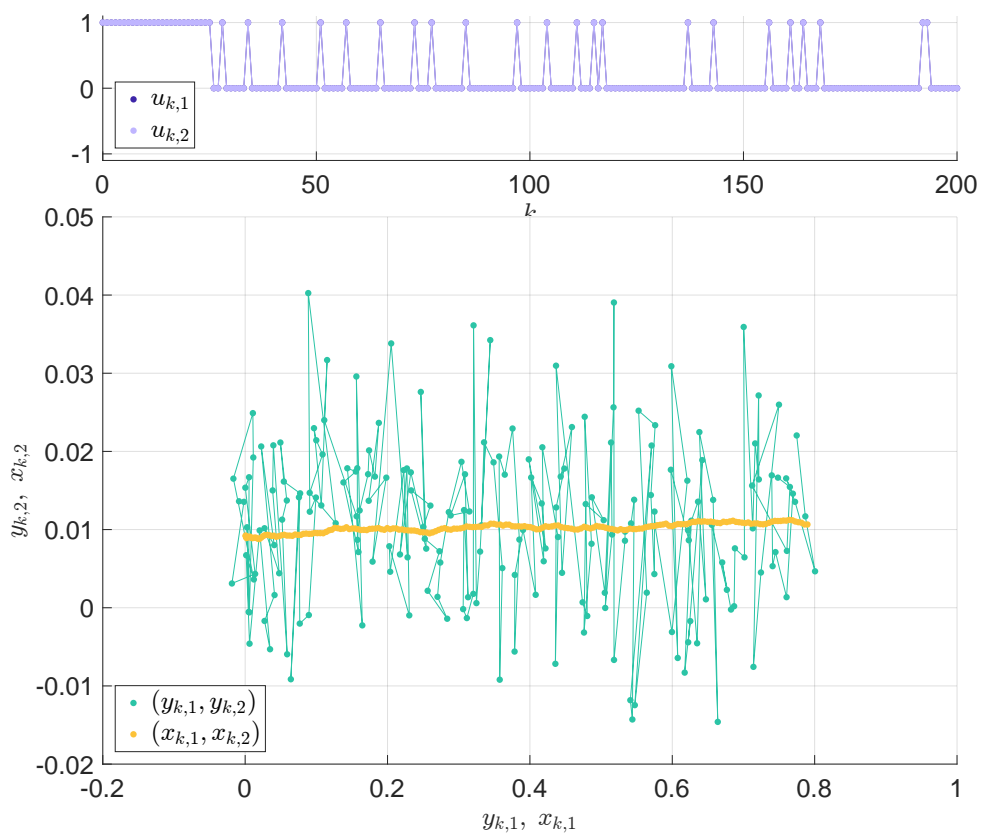
kaného hyperstavu (6.47), kdy je zapotřebí stanovit střední hodnotu Bellmanovy funkce pro budoucí hyperstav. Z dosažených výsledků je zároveň patrné, že algoritmus PF disponuje značně vyšší časovou náročností oproti algoritmu EKF, což může omezit jeho použití v aplikacích, kde je rychlost výpočtů rozhodující. Výpočetní nároky PF však lze ovlivnit vhodným výběrem počtu vzorků a způsobem převzorkování.



Obrázek 8.26: Typický průběh trajektorií vstupu \mathbf{u}_k a kartézských souřadnic spojité části stavu $(x_{k,1}, x_{k,2})$ a výstupu $(y_{k,1}, y_{k,2})$ mobilního robotu v případě pasivní detekce chyb.



Obrázek 8.27: Typický průběh trajektorií vstupu \mathbf{u}_k a kartézských souřadnic spojité části stavu $(x_{k,1}, x_{k,2})$ a výstupu $(y_{k,1}, y_{k,2})$ mobilního robotu v případě aktivní detekce chyb pomocí AFDR-EKF.



Obrázek 8.28: Typický průběh trajektorií vstupu \mathbf{u}_k a kartézských souřadnic spojité části stavu $(x_{k,1}, x_{k,2})$ a výstupu $(y_{k,1}, y_{k,2})$ mobilního robotu v případě aktivní detekce chyb pomocí AFDR-PF.

8.2 Detekce změn a chyb bez znalosti modelu

8.2.1 Systém prvního řádu

Tento příklad ověřuje a porovnává algoritmy detekce změn a chyb založené na datech, které byly prezentovány v kapitole 7. Konkrétně se jedná o adaptivní algoritmus 9 využívající vlastnosti nulového prostoru (ANUP) a adaptivní algoritmus 10 samoorganizujících se map (ASOM). Z důvodu předpokladů ANUP algoritmu a z důvodu snadné vizualizace jsou algoritmy porovnány na numerickém příkladu detekce změny chování lineárního skalárního systému.

Nechť je dán systém, který může být v každém časovém okamžiku $k = 0, 1, \dots$ popsán následujícím lineárním deterministickým modelem s následující strukturou

$$x_{k+1} = a_{\mu_k} x_k + b_{\mu_k} u_k, \quad (8.16)$$

kde $x_k \in \mathbb{R}$ je spojitá část stavu, $\mu_k \in \{1, 2\} = \mathcal{M}$ je index modelu, $u_k \in \mathbb{R}$ je vstup, a_{μ_k} spolu s b_{μ_k} jsou neznámé parametry pro všechny $\mu_k \in \mathcal{M}$. Pro účely simulace na konečném časovém horizontu 701 kroků se předpokládá vektor měření ve tvaru $\mathbf{z}_k = [x_k, x_{k-1}, u_{k-1}]^T$. Pro dvě možná chování systému jsou definovány následující parametry modelu

- $a_1 = 0.15, b_1 = 0.5,$
- $a_2 = 0.95, b_2 = 0.9.$

Chování systému se v čase vyvíjí v závislosti na indexu modelu, který je definován následovně

- $\mu_k = 1$ pro $k \in \{0, 1, \dots, 199\} \cup \{450, 451, \dots, 599\},$
- $\mu_k = 2$ pro $k \in \{200, 201, \dots, 449\} \cup \{600, 601, \dots, 700\}.$

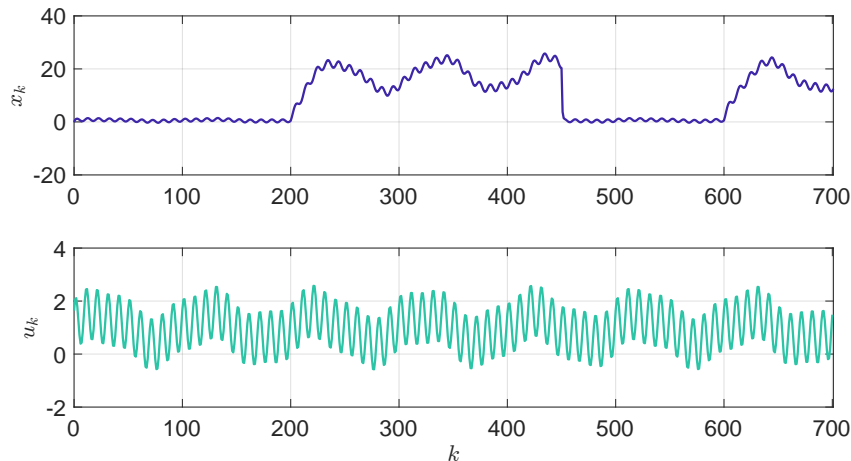
Generátor vstupního signálu je zvolen jako součet vážených funkcí sinus a konstanty

$$u_k = 1 + \sin(\pi k) + 0.5 \sin(0.02\pi k) + 0.15 \sin(0.048\pi k). \quad (8.17)$$

Počáteční stav je nulový, neboli $x_0 = 0$. Počáteční adaptivní počet modelů $N_\mu^a = 1$ a počáteční váha modelu $m_0^1 = 1$ jsou zvoleny pro oba algoritmy stejně. Mez tolerance ANUP algoritmu je nastavena na hodnotu $\epsilon_n = 1$. Pro ASOM algoritmus byly zvoleny následující parametry: $\epsilon_d = 12, \beta = 0.2, \kappa_0 = 1.5, \varsigma_0 = 1, \tau_\kappa = 10$ a $\tau_\varsigma = 15$. ANUP algoritmus vyžaduje stanovení pouze jednoho parametru oproti šesti parametrům v případě ASOM. Cílem úlohy je správně identifikovat model a detekovat změnu modelu.

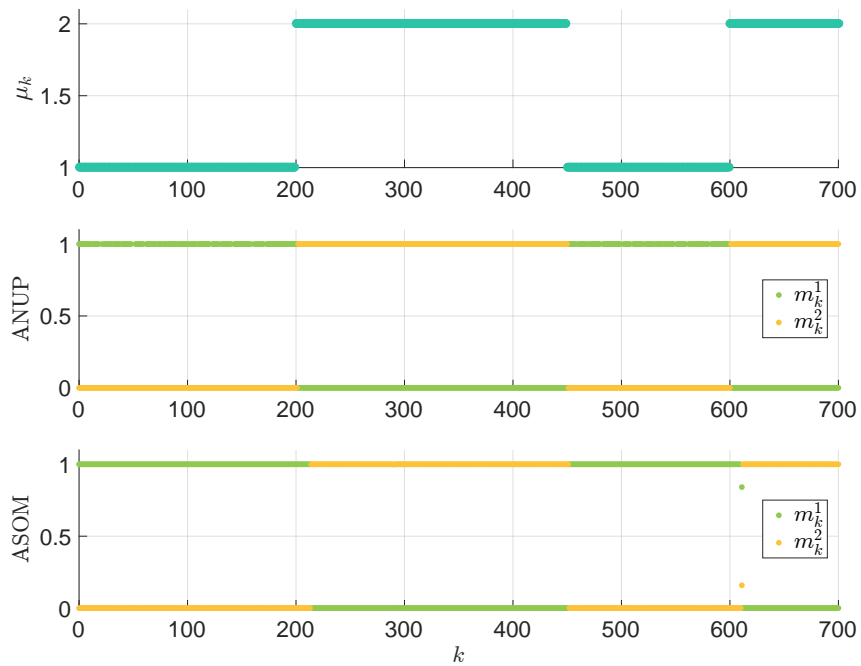
Trajektorie spojitě části stavu x_k a vstupu u_k na celém časovém horizontu jsou zobrazeny na obrázku 8.29. Z obrázku je patrné, že změny chování systému se výrazně projeví na hodnotách spojitě části stavu x_k , které jsou zároveň do jisté míry ovlivněny hodnotami vstupu u_k . Průběh trajektorie skutečného indexu modelu μ_k spolu s trajektorií vah modelů

m_k^i pro $i = 1, 2$ jsou prezentovány na obrázku 8.30. Při pohledu na hodnoty vah modelů m_k^i je zřejmé, že váhy v krátkém čase reagují na změnu indexu modelu μ_k a detekce změn probíhá korektně pro oba algoritmy. Hodnoty vah aktuálního indexu modelu se blíží hodnotě jedna, zatímco hodnoty druhého modelu se blíží nule.

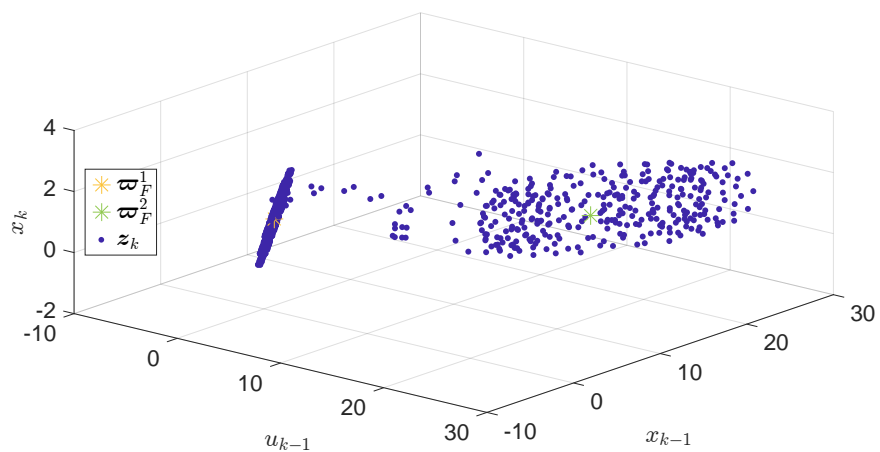


Obrázek 8.29: Trajektorie spojité části stavu x_k a vstupu u_k pro systém prvního řádu.

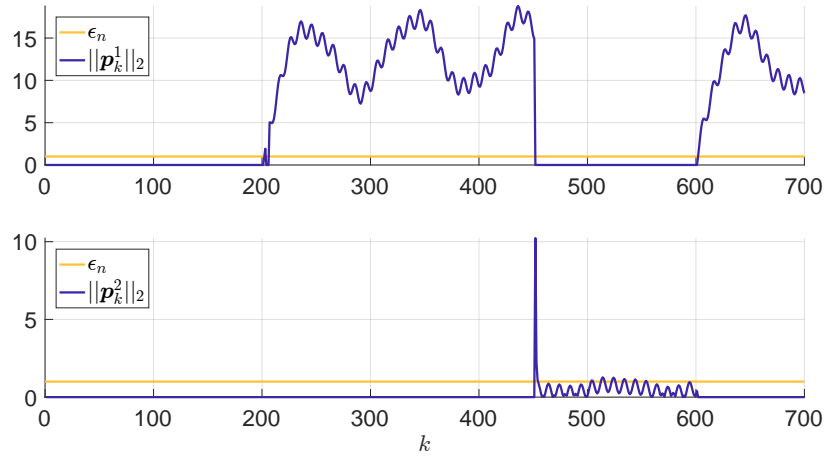
Jednotlivá měření \mathbf{z}_k v prostoru a váhy neuronů ϖ_F^1 a ϖ_F^2 počítané v ASOM algoritmu jsou porovnány na obrázku 8.31. Na obrázku jsou patrné dva oddělené shluky měření \mathbf{z}_k . Váhy obou neuronů ASOM algoritmu se ustálily v bodech, které se zdají být pomyslnými středy těchto shluků. Avšak v případě překrytí či blízkosti těchto regionů by ASOM algoritmus nejspíše nebyl schopný rozpoznat chování systému na základě vítězného neuronu. Možný scénář by představoval nežádoucí vytváření nových neuronů, a tedy i nových modelů chování. ASOM algoritmus tedy není vhodný na detekci změn chování, které se v měření projeví pouze nepatrně. Porovnání normy $\|\mathbf{p}_k^i\|_2$ s mezní hodnotou ϵ_n v případě ANUP algoritmu je zobrazeno na obrázku 8.32 a porovnání normy $\|\varpi_k^i - \mathbf{z}_k\|_2$ s mezní hodnotou ϵ_d v případě ASOM algoritmu je zobrazeno na obrázku 8.33. V případě ANUP je zřejmé, že hodnota normy $\|\mathbf{p}_k^i\|_2$ je velmi citlivá na změnu chování, jelikož hodnoty pro aktuální chování se blíží nule. To znamená, že mez tolerance ϵ_n by v tomto případě mohla být zvolena mnohem nižší. ANUP algoritmus lze na rozdíl od ASOM algoritmu použít i v případech, kdy se změny v chování systému projeví nepatrně. Z obrázku 8.33 je patrné, že mez tolerance ϵ_d v případě ASOM algoritmu je zvolena přesně tak, aby hodnoty $\|\varpi_k^i - \mathbf{z}_k\|_2$ pro aktuální chování byly těsně pod mezí. Ze simulačních výsledků je zřejmé, že oba algoritmy fungují v daném příkladu uspokojivě. Použití ANUP algoritmu je limitované třídou lineárních systémů, zatímco použití ASOM je limitované nutnou volbou mnoha parametrů, vektoru měření \mathbf{z}_k a samotnými vlastnostmi systému.



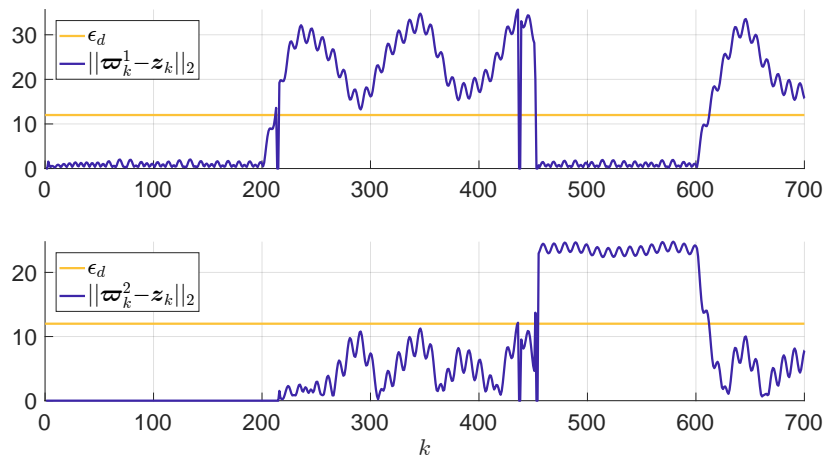
Obrázek 8.30: Trajektorie skutečného indexu modelu μ_k a trajektorie vah jednotlivých modelů m_k^i , $i = 1, 2$, pro ANUP a ASOM algoritmy.



Obrázek 8.31: Zobrazení dostupných měření \mathbf{z}_k v prostoru a vah neuronů \mathbf{w}_F^1 a \mathbf{w}_F^2 získaných na základě ASOM algoritmu.



Obrázek 8.32: Trajektorie signální normy $\|\mathbf{p}_k^i\|_2$ ANUP algoritmu pro $i = 1, 2$ a porovnání s mezí tolerance ϵ_n .



Obrázek 8.33: Trajektorie signální normy $\|\mathbf{w}_k^i - \mathbf{z}_k\|_2$ ASOM algoritmu pro $i = 1, 2$ a porovnání s mezí tolerance ϵ_d .

Kapitola 9

Závěr

Tato disertační práce se zabývala detekcí změn a chyb v dynamických systémech. Podle způsobu interakce mezi detektorem chyb a systémem lze rozlišovat mezi pasivní detekcí a aktivní detekcí chyb. V závislosti na způsobu modelování nejistot systému je dále možné rozlišovat mezi pravděpodobnostním přístupem a deterministickým přístupem k aktivní detekci chyb [Š9]. Tato práce pojednávala o pravděpodobnostním přístupu, kdy jsou nejistoty uvažovány ve formě náhodných procesů se známými pravděpodobnostními popisy. Chování systému bylo popsáno konečnou množinou známých modelů a cíl optimalizační úlohy spočíval v návrhu strategie, která minimalizuje vhodně zvolené kritérium kvality na nekonečném časovém horizontu. Formulovaná optimalizační úloha se řadí do třídy úloh s neúplnou informací o stavu. Řešení takových úloh je obvykle teoreticky i výpočetně složitější v porovnání s úlohami s úplnou informací o stavu, a proto byla původní úloha transformována na úlohu s úplnou informací o stavu, jejíž řešení bylo nalezeno dynamickým programováním. Jelikož nalezení exaktního řešení formulované úlohy je výpočetně i paměťově nereálné, bylo zapotřebí navrhnout vhodný aproximační postup návrhu aktivního detektoru chyb.

Úloha detekce změn a chyb bez znalosti modelu předpokládá, že model systému není znám předem a návrh pasivního detektoru chyb je proveden na základě dat. Takový způsob monitorování systému může pomoci snížit finanční i časové nároky vynaložené na tvorbu modelu. Jelikož v mnoha případech není jednoduché určit vztah mezi získanými daty a chováním systému, zdají se vhodným nástrojem řešení být metody strojového učení. V disertační práci byly navrženy dvě metody detekce změn a chyb bez znalosti modelu a zároveň bylo diskutováno jejich využití v úloze řízení bez znalosti modelu.

9.1 Shrnutí dosažených výsledků

Na základě přehledu aktuálního stavu problematiky byly v kapitole 3 identifikovány nedostatky současných pravděpodobnostních přístupů k aktivní detekci chyb a byly definovány následující cíle disertační práce:

Cíl 1: formulace úlohy návrhu aktivního detektoru chyb na nekonečném časovém horizontu,

Cíl 2: návrh řešení formulované úlohy,

Cíl 3: rozšíření formulace a řešení úlohy aktivní detekce chyb přidáním požadavků na chování systému,

Cíl 4: ilustrace vlastností navrženého řešení aktivní detekce chyb na numerických příkladech,

Cíl 5: návrh řešení úlohy detekce změn a chyb bez znalosti modelu založené na datech a jeho ověření na numerickém příkladě.

Ke splnění cílů byly využity poznatky z teorie pravděpodobnosti, teorie odhadu, teorie optimálního řízení a umělé inteligence. Dosažené výsledky výzkumu byly publikovány v předním odborném časopise oboru kybernetika a ve sbornících celosvětově uznávaných konferencí. Reference označené jako [Š ·] představují odkazy na autorovy publikace. Propojení cílů a dosažených výsledků je následující.

Cíl 1: Formulace úlohy návrhu aktivního detektoru chyb na nekonečném časovém horizontu vychází z předchozích výsledků obecné formulace úlohy aktivní detekce chyb a řízení na konečném časovém horizontu [15] a návrhu aktivního detektoru chyb s obecným návrhovým kritériem na nekonečném horizontu pro lineární gaussovské systémy s přímo měřitelnou spojitou částí stavu [77].

Disertační práce popisuje formulaci úlohy aktivní detekce chyb a řízení pro systémy, které mohou být popsány vícemodelovým přístupem se známými pravděpodobnostmi přechodu mezi modely. Model systému, aktivní detektor chyb a návrhové kritérium byly detailně popsány pro obecný případ nelineárního diskrétního stochastického systému s nepřímo měřitelným stavem. Na základě uvedené obecné formulace úlohy lze odvodit následující speciální případy, které byly představeny v níže uvedených publikacích:

- úloha s přímo měřitelnou spojitou částí stavu
 - lineární systém ([77]),
 - nelineární systém ([Š14]),
- úloha s nepřímo měřitelnou spojitou částí stavu
 - lineární systém ([Š8]),
 - nelineární systém ([Š3]).

Cíl 2: Návrh řešení formulované úlohy byl v disertační práci rozdělen do dvou na sebe navazujících částí. První část uvedená v kapitole 5 se zabývala přeformulováním úlohy s neúplnou informací o stavu na úlohu s úplnou informací o stavu. Druhá část uvedená v kapitole 6 se týkala samotného řešení transformované úlohy pomocí aproximačních metod dynamického programování a zpětnovazebního učení.

V obecném případě není spojitá ani diskrétní část stavu známá, proto byla původní úloha s neúplnou informací o stavu přeformulována na úlohu s úplnou informací o stavu využitím postačujících statistik stavu systému. Jelikož postačující statistiky nejsou příliš vhodné pro implementaci z důvodu jejich rozměrnosti, disertační práce představila vhodné formy aproximace postačujících statistik pomocí tzv. informačních statistik. Příkladem může být nahrazení hustoty pravděpodobnosti pomocí jejich prvních dvou momentů.

Nejjednodušší případ představuje úloha s přímo měřitelnou spojitou částí stavu, kdy je zapotřebí definovat pouze postačující statistiky pro diskrétní část stavu, tedy index modelu. V takovém případě je postačující statistikou podmíněná diskrétní hustota pravděpodobnosti, která může být vyjádřena využitím Bayesova vztahu [Š14]. U lineárních systémů s nepřímo měřitelnou spojitou částí stavu lze využít sadu KF a nějaký ze suboptimálních algoritmů redukcí zvyšující se výpočetní náročnost způsobenou exponenciálním růstem počtu možných posloupností indexů modelů. Algoritmus 1 založený na odhadu informačních statistik pomocí KF a ZPB byl publikován v [Š8]. Algoritmus 2 publikovaný v [Š3] představuje podobné řešení pro případ nelineárních gaussovských systémů s nepřímo měřitelnou spojitou částí stavu, jelikož k odhadu informačních statistik využívá EKF a ZPB. Vzhledem k tomu, že použití EKF pro systémy se silně nelineárními nebo nediferencovatelnými funkcemi systému není vhodné, lze v případě nelineárních negaussovských systémů s nepřímo měřitelnou spojitou částí stavu využít PF. Algoritmus 3 založený na bootstrap PF byl představen v [Š4].

K řešení přeformulované optimalizační úlohy aktivní detekce chyb s úplnou informací o stavu je nutné z důvodu snížení výpočetní složitosti použít aproximační postupy dynamického programování. Základní myšlenka spočívá ve vhodné reprezentaci Bellmanovy funkce, na základě které je počítán samotný aktivní detektor chyb. V disertační práci bylo představeno několik možných řešení vhodných pro různé úlohy, které se dají rozdělit do dvou skupin. První skupinu tvoří řešení založená na aproximaci Bellmanovy funkce pomocí tabulky, kdy chování dynamického systému může být modelováno jako Markovův rozhodovací proces. Transformace systému na Markovův rozhodovací proces spolu s adaptivním algoritmem 4 zobecněné iterace strategie byla prezentována v [Š11]. Možné řešení bez transformace systému představuje použití agregační funkce přímo na trajektorii informačních statistik ve formě hyperstavu a použití iteračních postupů dynamického programování. Porovnání algoritmu 5 iterace užitkové funkce s přímou kvantizací spojitého prostoru hyperstavů a algoritmu iterace strategie bylo diskutováno v [Š14]. Druhá skupina technik využívá k nahrazení Bellmanovy funkce lineární parametrickou nebo neparametrickou funkcionální aproximaci. Hledání řešení úlohy aktivní detekce chyb je inspirováno zpětnovazebním učením, a proto byl v disertační práci představen algoritmus 6 učení TD [Š5] a algoritmus 7 učení TDQ [Š6]. V disertační práci byl zároveň popsán algoritmus 8 využívající neparametrickou aproximaci získanou iterací užitkové funkce, který byl publikován v [Š8].

Cíl 3: Rozšíření formulace a řešení úlohy aktivní detekce chyb přidáním požadavků na chování systému je rozumné v případě jednotného návrhu detektoru chyb a regulátoru. Klí-

čový rozdíl spočívá ve volbě ztrátové funkce, která nově hodnotí kompromis mezi kvalitou detekce chyb a řízením. V závislosti na volbě návrhového koeficientu v kritériu je možné rozlišit mezi úlohou aktivní detekce chyb, úlohou aktivní detekce chyb a řízení a úlohou optimálního řízení.

Práce stručně popisuje formulaci úlohy návrhu aktivního detektoru chyb a regulátoru společně s přeformulováním na úlohu s úplnou informací o stavu a jejím řešením pomocí dynamického programování. Výsledky týkající se úlohy aktivní detekce chyb a řízení byly publikovány v [Š13] a [Š11].

Cíl 4: Vlastnosti navrženého řešení úlohy aktivní detekce chyb byly ukázány na třech různých numerických příkladech. V prvním příkladu byl uvažován kontinuálně míchaný bioreaktor, tzv. chemostat, který byl modelován jako dynamický systém s přímo měřitelnou spojitou částí stavu. Cílem úlohy byl společný návrh aktivního detektoru chyb a regulátoru, který minimalizuje pravděpodobnost chybného rozhodnutí a zároveň reguluje stav tak, aby co nejlépe sledoval konstantní referenční signál. Příklad obsahuje srovnání vlastností adaptivního algoritmu 4 s vlastnostmi neadaptivních řešení. Cílem druhého příkladu matematického kyvadla bylo srovnání různých metod řešení přeformulované optimalizační úlohy aktivní detekce chyb s úplnou informací o stavu. Vlastnosti algoritmu 6 učení TD, algoritmu 7 učení TDQ a algoritmu 8 využívajícího neparаметrickou aproximaci byly porovnány s pasivním detektorem chyb a aktivním detektorem chyb využívajícím generátor vstupního sinusového signálu. Poslední příklad týkající se návrhu aktivního detektoru chyb uvažoval nelineární systém mobilního robotu. Cílem příkladu bylo porovnat různé algoritmy přeformulování úlohy, konkrétně algoritmus 2 využívající EKF a ZPB a algoritmus 3 založený na bootstrap PF. Všechny numerické příklady vykazují nadějně výsledky navržených řešení.

Cíl 5: Metody detekce změn a chyb bez znalosti modelu využívající pouze měřená data lze využít v případě neznalosti modelu systému. Za předpokladu dostupnosti trénovacích dat lze využít některou z metod učení s učitelem, jako je například rozpoznání vzorů [43]. V případě neznámé závislosti mezi výstupními a vstupními daty je však nutné hledat jiné metody jako je analýza hlavních komponent [124] nebo samoorganizující se mapy [126].

Disertační práce představila dva algoritmy detekce změn a chyb založené na měřených datech. Algoritmus 9 využívá geometrické vlastnosti lineárních systémů. Rozhodnutí je založeno na rozdílnosti nulových prostorů nasbíraných dat pro různá chování systému. Druhý algoritmus využívá samoorganizující se mapy, neboli samoučící se neuronové sítě. K detekci změny dochází v důsledku rozdílnosti vstupního vektoru a vah jednotlivých neuronů tvořících výpočetní vrstvu neuronových sítí. Adaptivní algoritmus 10 samoorganizujících se map byl publikován v [Š1]. Tam byla zároveň ukázána možná aplikace pro úlohu optimálního řízení bez znalosti modelu. Ověření vlastností obou algoritmů bylo představeno na numerickém příkladě systému prvního řádu.

9.2 Možnosti budoucího výzkumu

Vzhledem k rostoucí složitosti systémů a technologickému trendu je v oblastech jako odhad stavu, automatické řízení nebo detekce chyb stále častěji skloňován výraz *vícerozměrové systémy*, neboli systémy, s nimiž nemůže být zacházeno pomocí standardních přístupů [143]. Speciální metody detekce chyb a řízení pro vícerozměrové systémy mohou být využity v kyberneticko-fyzikálních systémech nebo Internetu věcí z důvodu zvýšení efektivity, přehlednosti nebo odolnosti proti kybernetickým útokům. Oproti úloze odhadu stavu nebo řízení lze velmi těžko najít literaturu zabývající se detekcí změn a chyb ve vícerozměrových systémech, navíc se jedná výhradně o pasivní detekci chyb [144]–[147]. V případě aktivní detekce chyb byl k decentralizovanému řízení odolnému proti chybám nedávno použit deterministický přístup [148]. Avšak podle nejlepšího vědomí autora nebyl stochastický přístup k aktivní detekci chyb pro vícerozměrové systémy dodnes uvažován. Možným směrem budoucího výzkumu by mohlo být vytvoření nového rámce aktivní detekce chyb a řízení pro stochastické vícerozměrové systémy zahrnující formulaci úlohy a její řešení.

Dalším možným směrem budoucího výzkumu by mohla být analýza konvergence uvedených algoritmů aktivní detekce chyb a řízení, která by pomohla dále zvýšit kvalitu detekce chyb a zároveň snížit bariéru mezi teoretickým výzkumem a reálnou aplikací metod.

Literatura

- [1] M. Basseville a I. V. Nikiforov, *Detection of Abrupt Changes - Theory and Application*. Englewood Cliffs, NJ, USA: Prentice Hall, 1993.
- [2] L. H. Chiang, E. L. Russel a R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. London, United Kingdom: Springer Verlag, 2001.
- [3] X. J. Zhang, *Auxiliary Signal Design in Fault Detection and Diagnosis*. Heidelberg, Germany: Springer Verlag, 1989.
- [4] S. L. Campbell a R. Nikoukhah, *Auxiliary Signal Design for Failure Detection*. Princeton, NJ, USA: Princeton University Press, 2004.
- [5] L. Ljung, *System Identification: Theory for the User*. Englewood Cliffs, NJ, USA: Prentice Hall, 1987.
- [6] A. C. Atkinson, A. N. Donev a R. D. Tobias, *Optimum Experimental Designs, with SAS*. New York, NY, USA: Oxford University Press, 2007.
- [7] N. M. Filatov a H. Unbehauen, *Adaptive Dual Control: Theory and Applications*. Berlin, Germany: Springer Verlag, 2004.
- [8] A. E. Ashari, R. Nikoukhah a S. L. Campbell, “Asymptotic behavior and solution approximation of active robust fault detection for closed-loop systems”, in *Proceedings of the IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, P.R. China, 2009, s. 1026–1031.
- [9] G. R. Marseglia, J. K. Scott, L. Magni, R. D. Braatz a D. M. Raimondo, “A hybrid stochastic-deterministic approach for active fault diagnosis using scenario optimization”, in *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, 2014, s. 1102–1107.
- [10] J. A. Paulson, D. M. Raimondo, R. Findeisen, R. D. Braatz a S. Streif, “Guaranteed active fault diagnosis for uncertain nonlinear systems”, in *Proceedings of the 2014 European Control Conference*, Strasbourg, France, 2014, s. 926–931.
- [11] J. K. Scott, R. Findeisen, R. D. Braatz a D. M. Raimondo, “Input design for guaranteed fault diagnosis using zonotopes”, *Automatica*, sv. 50, č. 6, s. 1580–1589, červ. 2014.

- [12] J. Wang, W. Ge, H. Wu a J. Zhou, “Active fault detection based on residual ellipsoid”, in *Proceedings of the 35th Chinese Control Conference*, Chengdu, China, 2016, s. 6784–6789.
- [13] F. Kerestecioglu, *Change detection and input design in dynamical systems*. Taunton, United Kingdom: Research Studies Press, 1993.
- [14] L. Blackmore, S. Rajamanoharan a B. C. Williams, “Active estimation for jump Markov linear systems”, *IEEE Transactions on Automatic Control*, sv. 53, č. 10, s. 2223–2236, lis. 2008.
- [15] M. Šimandl a I. Punčochář, “Active fault detection and control: unified formulation and optimal design”, *Automatica*, sv. 45, č. 9, s. 2052–2059, zář. 2009.
- [16] A. Mesbah, S. Streif, R. Findeisen a R. D. Braatz, “Active fault diagnosis for nonlinear systems with probabilistic uncertainties”, in *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, 2014, s. 7079–7084.
- [17] K. P. Murphy, “Introduction”, in *Machine Learning: A Probabilistic Perspective*, Cambridge, MA, USA: MIT Press, 2012, kap. 1, s. 1–25.
- [18] T. Pultarova, “Exomars crash: is European rover’s future uncertain?”, *Engineering & Technology*, sv. 11, č. 11, s. 17–17, 2016.
- [19] S. Jesudas, *Failure analysis of Samsung Note 7*, Seoul, Korea: UL, 2017.
- [20] M. Blanke, M. Kinnaert, J. Lunze a M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, 3rd. Heidelberg, Germany: Springer-Verlag, 2016.
- [21] R. Isermann, *Fault-Diagnosis Applications*. Heidelberg, Germany: Springer, 2011.
- [22] X. Dai a Z. Gao, “From model, signal to knowledge: a data-driven perspective of fault detection and diagnosis”, *IEEE Transactions on Industrial Informatics*, sv. 9, č. 4, s. 2226–2238, lis. 2013.
- [23] S. Simani, C. Fantuzzi a R. J. Patton, *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. London, United Kingdom: Springer-Verlag, 2003.
- [24] S. X. Ding, *Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms and Tools*, 2nd. London, United Kingdom: Springer, 2013.
- [25] K. J. Åström a R. M. Murray, *Feedback Systems*. Princeton, NJ, USA: Princeton University Press, 2008.
- [26] V. Palade, C. D. Bocaniala a L. C. Jain, *Computational Intelligence in Fault Diagnosis*. London, United Kingdom: Springer Verlag, 2006.
- [27] M. Witczak, *Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems: From Analytical to Soft Computing Approaches*. Heidelberg, Germany: Springer Verlag, 2007.

- [28] J.-S. R. Jang, C.-T. Sun a E. Mizutani, *Neuro-Fuzzy And Soft Computing Jang: A Computational Approach to Learning and Machine Intelligence*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1997.
- [29] R. J. Patton, P. M. Frank a R. N. Clarke, *Issues of Fault Diagnosis for Dynamic Systems*. London, United Kingdom: Springer Verlag, 2000.
- [30] J. Korbicz, J. M. Koscielny, Z. Kowalczyk a W. Cholewa, *Fault Diagnosis: Models, Artificial Intelligence, Applications*. Heidelberg, Germany: Springer, 2004.
- [31] R. Isermann, *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*. Heidelberg, Germany: Springer Verlag, 2006.
- [32] G. Vachtsevanos, F. L. Lewis, M. Roemer, A. Hess a B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2006.
- [33] J. Ma a J. Jiang, “Applications of fault detection and diagnosis methods in nuclear power plants: a review”, *Progress in Nuclear Energy*, sv. 53, č. 3, s. 255–266, led. 2011.
- [34] A. Zolghadri, “Advanced model-based fdir techniques for aerospace systems: today challenges and opportunities”, *Progress in Aerospace Sciences*, sv. 53, č. 3, s. 18–29, dub. 2012.
- [35] A. Gheorghe, A. Zolghadri, J. Cieslak, P. Goupil, R. Dayre a H. Le Berre, “Model-based approaches for fast and robust fault detection in an aircraft control surface servo loop: from theory to flight tests”, *IEEE Control Systems*, sv. 33, č. 3, s. 20–84, červ. 2013.
- [36] P. Pisu, “Fault detection and isolation with applications to vehicle systems”, in *Automotive Air Conditioning: Optimization, Control and Diagnosis*, Cham, Switzerland: Springer, 2016, kap. 13, s. 293–321.
- [37] Z. Gao, C. Cecati a S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniques - part I: fault diagnosis with model-based and signal-based approaches”, *IEEE Transactions on Industrial Electronics*, sv. 62, č. 6, s. 3757–3767, červ. 2015.
- [38] —, “A survey of fault diagnosis and fault-tolerant techniques — part II : fault diagnosis with knowledge-based and hybrid/active approaches”, *IEEE Transactions on Industrial Electronics*, sv. 62, č. 6, s. 3768–3774, červ. 2015.
- [39] R. S. Mangoubi, *Robust Estimation and Failure Detection: A Concise Treatment*. London, United Kingdom: Springer-Verlag, 1998.
- [40] J. Chen a R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Boston, MA, USA: Springer, 1999.
- [41] A. Wald, *Sequential Analysis*. Dover Publications, 1947.
- [42] M. Basseville, “Distance measures for signal processing and pattern recognition”, *Signal processing*, sv. 18, č. 4, s. 349–369, červ. 1989.

- [43] R. O. Duda, P. E. Hart a D. G. Stork, *Pattern Classification*, 2nd. New York, NY, USA: John Wiley & Sons, Inc., 2000.
- [44] L. Blackmore a B. Williams, “Finite horizon control design for optimal model discrimination”, in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, Seville, Spain, 2005, s. 3795–3802.
- [45] —, “Finite horizon control design for optimal discrimination between several models”, in *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, 2006, s. 1147–1152.
- [46] J. Nocedal a S. J. Wright, *Numerical Optimization*, 2nd. New York, NY, USA: Springer Verlag, 2006.
- [47] M. Šimandl a P. Herejt, “Information processing strategies and multiple model fault detection”, in *Proceedings of the 22nd IASTED International Conference Modeling, Identification, and Control*, Calgary, Canada, 2003, s. 347–352.
- [48] M. Šimandl, I. Punčochář a P. Herejt, “Optimal input and decision in multiple model fault detection”, in *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, 2005, s. 1877–1877.
- [49] R. E. Bellman, *Dynamic Programming*, 1. vyd. Princeton, NJ, USA: Princeton University Press, 1957.
- [50] D. P. Bertsekas, *Dynamic Programming and Optimal Control (Volume I)*, 2nd. Belmont, MA, USA: Athena Scientific, 2000.
- [51] I. Punčochář, J. Široký a M. Šimandl, “Constrained active fault detection and control”, *IEEE Transactions on Automatic Control*, sv. 60, č. 1, s. 253–258, led. 2015.
- [52] C. N. Nett, “Algebraic aspects of linear control system stability”, *IEEE Transactions on Automatic Control*, sv. 31, č. 10, s. 941–949, říj. 1986.
- [53] M. L. Tyler a M. Morari, “Optimal and robust design of integrated control and diagnostic modules”, in *Proceedings of the 1994 American Control Conference*, Baltimore, MD, USA, 1994, s. 2060–2064.
- [54] J. Stoustrup, M. J. Grimble a H. Niemann, “Design of integrated systems for the control and detection of actuator/sensor faults”, *Sensor Review*, sv. 17, č. 2, s. 138–149, 1997.
- [55] V. Kučera, “A method to teach the parameterization of all stabilizing controllers”, in *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011, s. 6355–6360.
- [56] M. Khosrowjerdi, R. Nikoukhah a N. Safari-Shad, “A mixed $\mathcal{H}_2/\mathcal{H}_\infty$ approach to simultaneous fault detection and control”, *Automatica*, sv. 40, č. 2, s. 261–267, zář. 2004.
- [57] H. Niemann a N. K. Poulsen, “Active fault diagnosis in closed-loop systems”, in *Proceedings of the 16th IFAC World Congress*, sv. 16, Prague, Czech Republic, 2005.

- [58] H. Niemann, “A setup for active fault diagnosis”, *IEEE Transactions on Automatic Control*, sv. 51, č. 9, s. 1572–1578, zář. 2006.
- [59] A. Sekunda, H. Niemann a N. K. Poulsen, “Active fault detection based on a statistical test”, in *Proceedings of the 3rd International Conference on Control and Fault-Tolerant Systems (SysTol)*, Barcelona, Spain, 2016, s. 511–518.
- [60] J. Wang, J. Zhang, B. Qu, H. Wu a J. Zhou, “Unified architecture of active fault detection and partial active fault-tolerant control for incipient faults”, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, sv. 47, č. 7, s. 1688–1700, čvc 2017.
- [61] R. Nikoukhah, “Guaranteed active failure detection and isolation for linear dynamical systems”, *Automatica*, sv. 34, č. 11, s. 1345–1358, lis. 1998.
- [62] R. Nikoukhah, S. L. Campbell, K. G. Horton a F. Delebecque, “Auxiliary signal design for robust multimodel identification”, *IEEE Transactions on Automatic Control*, sv. 47, č. 1, s. 158–164, led. 2002.
- [63] I. Andjelkovic, K. Sweetingham a S. L. Campbell, “Active fault detection in nonlinear systems using auxiliary signals”, in *2008 American Control Conference*, Seattle, WA, USA, 2008, s. 2142–2147.
- [64] S. Zhai, W. Wang a H. Ye, “Auxiliary signal design for active fault detection based on set-membership”, in *Proceedings of the 9th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Paris, France, 2015, s. 452–457.
- [65] J. K. Scott, R. Findeisen, R. D. Braatz a D. M. Raimondo, “Design of active inputs for set-based fault diagnosis”, in *Proceedings of the 2013 American Control Conference*, Washington, DC, USA, 2013, s. 3561–3566.
- [66] D. M. Raimondo, G. Roberto Marseglia, R. D. Braatz a J. K. Scott, “Closed-loop input design for guaranteed fault diagnosis using set-valued observers”, *Automatica*, sv. 74, s. 107–117, pros. 2016.
- [67] S. M. Tabatabaeipour, “Active fault detection and isolation of discrete-time linear time-varying systems: a set-membership approach”, *International Journal of Systems Science*, sv. 46, č. 11, s. 1917–1933, říj. 2013.
- [68] M. Šimandl, I. Punčochář a J. Královec, “Rolling horizon for active fault detection”, in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, Seville, Spain, 2005, s. 3789–3794.
- [69] M. Šimandl a I. Punčochář, “Closed loop information processing strategy for optimal fault detection and control”, in *Proceedings of the 14th IFAC Symposium on System Identification (SYSID)*, Newcastle, Australia, 2006, s. 487–492.
- [70] J. Široký, M. Šimandl, D. Axehill a I. Punčochář, “An optimization approach to resolve the competing aims of active fault detection and control”, in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, FL, USA, 2011, s. 3712–3717.

- [71] I. Punčochář, “Optimal active change detection and control of stochastic systems”, Disertační práce, Faculty of Applied Sciences, University of West Bohemia, 2008.
- [72] J. Široký, “Active fault detection and control”, Disertační práce, Faculty of Applied Sciences, University of West Bohemia, 2012.
- [73] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge, United Kingdom: Cambridge University Press, 2013.
- [74] M. Šimandl, J. Široký a I. Punčochář, “New special cases of general active change detection and control problem”, in *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011, s. 4260–4265.
- [75] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
- [76] I. Punčochář a M. Šimandl, “Infinite horizon input signal for active fault detection in controlled Markov chains”, in *Proceedings of the IASTED International Conference Intelligent systems and Control (ISC 2013)*, Marina del Rey, USA, 2013, s. 807–028.
- [77] I. Punčochář a M. Šimandl, “On the infinite horizon active fault detection problem for linear Gaussian multiple models”, in *Proceedings of the 2013 Conference on Control and Fault-Tolerant Systems (SysTol)*, Nice, France, 2013, s. 184–189.
- [78] J. Široký, I. Punčochář a M. Šimandl, “Active fault detection and constrained control of air handling unit”, in *Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Mexico City, Mexico, 2012.
- [79] D. Vrabie, K. G. Vamvoudakis a F. L. Lewis, *Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles*. London, United Kingdom: IET, 2013.
- [80] G. B. Dantzig a M. N. Thapa, *Linear Programming: Introduction*. New York, NY, USA: Springer-Verlag, 1997.
- [81] D. P. Bertsekas, *Nonlinear programming*, 2nd. Belmont, MA, USA: Athena Scientific, 1999.
- [82] J. Lee a S. Leyffer, *Mixed Integer Nonlinear Programming*. New York, NY, USA: Springer, 2012.
- [83] A. Prékopa, *Stochastic Programming*. Dodrecht, The Netherlands: Springer, 1995.
- [84] D. P. Bertsekas, *Dynamic Programming and Optimal Control (Volume II)*, 2nd. Belmont, MA, USA: Athena Scientific, 2001.
- [85] D. P. Bertsekas a J. N. Tsitsiklis, *Introduction to Probability*, 2nd. Belmont, MA, USA: Athena Scientific, 2008.

- [86] K. Watanabe a S. G. Tzafestas, “Generalized pseudo-Bayes estimation and detection for abruptly changing systems”, *Journal of Intelligent and Robotic Systems*, sv. 7, č. 1, s. 95–112, ún. 1993.
- [87] H. A. P. Blom a Y. Bar-Shalom, “Interacting multiple model algorithm for systems with Markovian switching coefficients.”, *IEEE Transactions on Automatic Control*, sv. 33, č. 8, s. 780–783, srp. 1988.
- [88] N. J. Gordon, D. Salmond a A. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation”, *IEE Proceedings F - Radar and Signal Processing*, sv. 140, č. 2, s. 107–113, dub. 1993.
- [89] J. D. Hol, T. B. Schön a F. Gustafsson, “On resampling algorithms for particle filters”, in *Proceedings of the 2006 IEEE Nonlinear Statistical Signal Processing Workshop*, Cambridge, United Kingdom, 2006, s. 79–82.
- [90] A. Doucet, N. De Freitas a N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer-Verlag, 2001.
- [91] O. Straka a M. Šimandl, “A survey of sample size adaptation techniques for particle filters”, in *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, 2009, s. 1358–1363.
- [92] H. Driessen a Y. Boers, “Efficient particle filter for jump Markov nonlinear systems”, *IEE Proceedings Journals & Magazines Radar, Sonar and Navigation*, sv. 152, č. 5, s. 323–326, říj. 2005.
- [93] R. S. Sutton a A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 1998.
- [94] E. V. Denardo, *Dynamic Programming Models and Applications*. Mineola, NY, USA: Dover Publications, 2003.
- [95] D. P. Bertsekas a J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA, USA: Athena Scientific, 1996.
- [96] F. L. Lewis a D. Vrabie, “Reinforcement learning and adaptive dynamic programming for feedback control”, *Circuits and Systems Magazine, IEEE*, s. 40–58, srp. 2009.
- [97] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, 1st. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007.
- [98] D. G. Luenberger, *Introduction to Dynamic Systems: Theory, Models, and Applications*. New York, NY, USA: John Wiley & Sons, Inc., 1979.
- [99] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2005.
- [100] E. Ikonen a U. Kortela, “Adaptive process control using controlled finite Markov chains based on multiple models”, in *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, 2008, s. 7919–7924.

- [101] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 1994.
- [102] ———, “Dynamic programming”, in *Encyclopedia of Physical Science and Technology*, 3rd, Academic Press, 2002, s. 673–696.
- [103] M. L. Puterman a M. C. Shin, “Modified policy iteration algorithms for discounted Markov decision problems”, *Management Science*, sv. 24, č. 11, s. 1127–1137, čvc 1978.
- [104] S. J. Julier a J. K. Uhlmann, “A new extension of the Kalman filter to nonlinear systems”, in *Proceedings of the SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI*, Orlando, FL, USA, 1997, s. 182–193.
- [105] R. S. Sutton, “Learning to predict by the methods of temporal differences”, *Machine Learning*, sv. 3, s. 9–44, srp. 1988.
- [106] B. Van Roy, “Neuro-dynamic programming: overview and recent trends”, in *Handbook of Markov Decision Processes: Methods and Applications*, E. A. Feinberg a A. Shwartz, ed., 1st, New York, NY, USA: Springer, 2002, kap. 14, s. 431–459.
- [107] G. Tesauro, “Practical issues in temporal difference learning”, *Machine Learning*, sv. 8, č. 3-4, s. 257–277, květ. 1992.
- [108] ———, “Temporal difference learning and TD-Gammon”, *Communications of the ACM*, sv. 38, č. 3, s. 58–68, břez. 1995.
- [109] A. L. Samuel, “Some studies in machine learning using the game of checkers”, *IBM Journal of Research and Development*, sv. 3, č. 3, s. 210–229, čvc 1959.
- [110] J. Baxter, A. Tridgell a L. Weaver, “Learning to play chess using temporal differences”, *Machine Learning*, sv. 40, č. 3, s. 243–263, zář. 2000.
- [111] L. Buşoniu, R. Babuška, B. D. Schutter a D. Ernst, *Reinforcement learning and dynamic programming using function approximators*. Boca Raton, FL, USA: CRC Press, 2010.
- [112] C. J. C. H. Watkins a P. Dayan, “Q-learning”, *Machine Learning*, sv. 8, č. 3, s. 279–292, 1992.
- [113] A. J. Smola a B. Schölkopf, “A tutorial on support vector regression”, *Statistics and Computing*, sv. 14, s. 199–222, srp. 2004.
- [114] W. B. Powell, “Approximating value functions”, in *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, 2nd, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007, kap. 8, s. 225–269.
- [115] M. P. Deisenroth, J. Peters a C. E. Rasmussen, “Approximate dynamic programming with Gaussian processes”, in *Proceedings of the 2008 American Control Conference*, Seattle, WA, USA, červ. 2008, s. 4480–4485.
- [116] M. P. Deisenroth, C. E. Rasmussen a J. Peters, “Gaussian process dynamic programming”, *Neurocomputing*, sv. 72, č. 7-9, s. 1508–1524, břez. 2009.

- [117] B. Bethke a J. P. How, “Approximate dynamic programming using Bellman residual elimination and Gaussian process regression”, in *Proceedings of the 2009 American Control Conference*, St. Louis, MO, USA, 2009, s. 745–750.
- [118] A. George, W. B. Powell a S. R. Kulkarni, “Value function approximation using multiple aggregation for multiattribute resource management”, *Journal of Machine Learning Research*, sv. 9, s. 2079–2111, říj. 2008.
- [119] J. M. Lee, N. S. Kaisare a J. H. Lee, “Choice of approximator and design of penalty function for an approximate dynamic programming based control approach”, *Journal of Process Control*, sv. 16, č. 2, s. 135–156, ún. 2006.
- [120] B. Kiumarsi, F. L. Lewis a D. S. Levine, “Optimal control of nonlinear discrete time-varying systems using a new neural network approximation structure”, *Neurocomputing*, sv. 156, s. 157–165, květ. 2015.
- [121] T. Söderström a P. Stoica, *System Identification*. Englewood Cliffs, NJ, USA: Prentice Hall, 1989.
- [122] X. R. Li a Y. Zhang, “Multiple-model estimation with variable structure part v: likely-model set algorithm”, *IEEE Transactions on Aerospace and Electronic Systems*, sv. 36, č. 2, s. 448–466, lis. 2000.
- [123] M. Cord a P. Cunningham, *Machine Learning Techniques for Multimedia*. Heidelberg, Germany: Springer-Verlag, 2008.
- [124] L. H. Chiang, E. L. Russell a R. D. Braatz, “Fault diagnosis in chemical processes using fisher discriminant analysis, discriminant partial least squares, and principal component analysis”, *Chemometrics and Intelligent Laboratory Systems*, sv. 50, č. 2, s. 243–252, břez. 2000.
- [125] T. Kohonen, “Self-organized formation of topologically correct feature maps”, *Biological Cybernetics*, sv. 43, č. 1, s. 59–69, led. 1982.
- [126] —, *Self-Organizing Maps*, 3rd. Heidelberg, Germany: Springer-Verlag, 2001.
- [127] B. Fritzke, “A growing neural gas network learns topologies”, *Advances in Neural Information Processing Systems 7*, sv. 7, č. 1, s. 625–632, břez. 1995.
- [128] —, “Growing cell structures—a self-organizing network for unsupervised and supervised learning”, *Neural Networks*, sv. 7, č. 9, s. 1441–1460, břez. 1994.
- [129] F. Takens, “Detecting strange attractors in turbulence”, *Dynamical Systems and Turbulence, Lecture Notes in Mathematics*, sv. 898, s. 366–381, 1981.
- [130] H. Sang, A. E. Gelfand, C. Lennard, G. Hegerl a B. Hewitson, “Interpreting self-organizing maps through space-time data models”, *The Annals of Applied Statistics*, sv. 2, č. 4, s. 1194–1216, pros. 2008.
- [131] G. H. Golub a C. F. Van Loan, *Matrix computations*, 4th. Baltimore, MD, USA: The Johns Hopkins University Press, 2012.

- [132] F. L. Lewis a V. L. Syrmos, *Optimal Control*, 3rd. New York, NY, USA: John Wiley & Sons, Inc., 1995.
- [133] B. Kiumarsi a F. L. Lewis, “Actor-critic-based optimal tracking for partially unknown nonlinear discrete-time systems”, *IEEE Transactions on Neural Networks and Learning Systems*, sv. 26, č. 1, s. 140–151, led. 2015.
- [134] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour a M.-B. Naghibi-Sistani, “Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics”, *Automatica*, sv. 50, č. 4, s. 1167–1175, dub. 2014.
- [135] B. Kiumarsi, H. Modares, F. L. Lewis a Z.-P. Jiang, “ \mathcal{H}_∞ optimal control of unknown linear discrete-time systems: an off-policy reinforcement learning approach”, in *Proceedings of the 7th IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and Robotics, Automation and Mechatronics (RAM)*, Siem Reap, Cambodia, 2015, s. 41–46.
- [136] Y. Jiang a Z.-P. Jiang, “Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics”, *Automatica*, sv. 48, č. 10, s. 2699–2704, říj. 2012.
- [137] H. Li, D. Liu a D. Wang, “Integral reinforcement learning for linear continuous-time zero-sum games with completely unknown dynamics”, *IEEE Transactions on Automation Science and Engineering*, sv. 11, č. 3, s. 706–714, čvc 2014.
- [138] H. Modares, F. L. Lewis a Z. P. Jiang, “ \mathcal{H}_∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning”, *IEEE Transactions on Neural Networks and Learning Systems*, sv. 26, č. 10, s. 2550–2562, říj. 2015.
- [139] D. E. Seborg, T. F. Edgar a D. A. Mellichamp, *Process Dynamics and Control*, 2nd. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2004.
- [140] F. Garces, V. M. Becerra, C. Kambhampati a K. Warwick, *Strategies for Feedback Linearisation*. London, United Kingdom: Springer Verlag, 2003.
- [141] B. Efron a R. J. Tibshirani, *An Introduction to the Bootstrap*. New York, NY, USA: Chapman & Hall/CRC, 1994.
- [142] R. Fierro a F. L. Lewis, “Control of a nonholonomic mobile robot: backstepping kinematics into dynamics”, in *Proceedings of the 34th IEEE Conference on Decision and Control (CDC)*, New Orleans, LA, USA, 1995, s. 3805–3810.
- [143] L. Bakule a J. Lunze, “Decentralized design of feedback control large-scale systems”, *Kybernetika*, sv. 24, s. 1–100, 1988.
- [144] W. H. Chung a J. L. Speyer, “A decentralized fault detection filter”, in *Proceedings of the 1998 American Control Conference*, Philadelphia, PA, USA, 1998, s. 2017–2021.
- [145] M. G. Singh, M. F. Hassan, Y. L. Chen, D. S. Li a Q. R. Pan, “New approach to failure detection in large-scale systems”, *IEE Proceedings of Control Theory and Applications*, sv. 130, č. 5, s. 243–249, zář. 1983.

- [146] R. M. Ferrari, T. Parisini a M. M. Polycarpou, “Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: an adaptive approximation approach”, *IEEE Transactions on Automatic Control*, sv. 57, č. 2, s. 275–290, ún. 2012.
- [147] A. Teixeira, I. Shames, H. Sandberg a K. H. Johansson, “Distributed fault detection and isolation resilient to network model uncertainties”, *IEEE Transactions on Cybernetics*, sv. 44, č. 11, s. 2024–2037, lis. 2014.
- [148] D. M. Raimondo, F. Boem, A. Gallo a T. Parisini, “A decentralized fault-tolerant control scheme based on active fault diagnosis”, in *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, Las Vegas, NV, USA, 2016, s. 2164–2169.

Seznam publikací autora

Publikace související s detekcí změn a chyb

- [Š1] J. Škach, B. Kiumarsi, F. L. Lewis a O. Straka, “Actor-critic off-policy learning for optimal control of multiple-model discrete-time systems”, *Přijato k publikaci v časopise IEEE Transactions on Cybernetics*, s. 1–12, 2016.
- [Š2] J. Škach, O. Straka a I. Punčochář, “Efficient active fault diagnosis using adaptive particle filter”, in *Přijato k publikaci ve sborníku Proceedings of the 56th IEEE Conference on Decision and Control (CDC)*, Melbourne, Australia, 2017.
- [Š3] J. Škach a I. Punčochář, “Input design for fault detection using extended Kalman filter and reinforcement learning”, in *Proceedings of the 20th IFAC World Congress*, Toulouse, France, 2017, s. 7563–7568.
- [Š4] J. Škach, I. Punčochář a O. Straka, “Active fault diagnosis for jump Markov nonlinear systems”, in *Proceedings of the 20th IFAC World Congress*, Toulouse, France, 2017, s. 7569–7574.
- [Š5] J. Škach, I. Punčochář a F. L. Lewis, “Optimal active fault diagnosis by temporal-difference learning”, in *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, Las Vegas, NV, USA, 2016, s. 2146–2151.
- [Š6] —, “Temporal-difference Q-learning in active fault diagnosis”, in *Proceedings of the 3rd International Conference on Control and Fault-Tolerant Systems (SysTol)*, Barcelona, Spain, 2016, s. 281–286.
- [Š7] J. Škach, “Návrh pokročilých algoritmů v úloze aktivní detekce chyb”, in *SVK FAV Studentská vědecká konference: Sborník rozšířených abstraktů*, Pilsen, Czech Republic: University of West Bohemia, 2016, s. 121–122.
- [Š8] I. Punčochář, J. Škach a M. Šimandl, “Infinite time horizon active fault diagnosis based on approximate dynamic programming”, in *Proceedings of the 54th IEEE Conference on Decision and Control (CDC)*, Osaka, Japan, 2015, s. 4456–4461.
- [Š9] J. Škach a I. Punčochář, “Active fault detection: a comparison of probabilistic methods”, *Journal of Physics: Conference Series*, sv. 659, č. 1, s. 1–12, 2015.
- [Š10] J. Škach, *Detekce změn a chyb v dynamických systémech*, Teze k disertační práci, Pilsen, Czech Republic: University of West Bohemia, 2015.

- [Š11] I. Punčochář, J. Škach a M. Šimandl, “Adaptive generalized policy iteration in active fault detection and control”, in *Proceedings of the 9th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Paris, France, 2015, s. 505–510.
- [Š12] J. Škach, “Aktivní detekce chyb a řízení”, in *SVK FAV Studentská vědecká konference: Sborník rozšířených abstraktů*, Pilsen, Czech Republic: University of West Bohemia, 2015, s. 115–116.
- [Š13] J. Škach, I. Punčochář a M. Šimandl, “Approximate active fault detection and control”, *Journal of Physics: Conference Series*, sv. 570, č. 1, s. 1–9, 2014.
- [Š14] M. Šimandl, J. Škach a I. Punčochář, “Approximation methods for optimal active fault detection”, in *Proceedings of the 22nd Mediterranean Conference on Control and Automation*, Palermo, Italy, 2014, s. 103–108.

Ostatní publikace

- [ŠO1] M. Šimandl, I. Punčochář, O. Straka, L. Král a J. Škach, “Analýza rizik a metody pro bezpečnou lokalizaci na bázi GNSS a digitální mapy tratě”, AŽD Praha s.r.o., tech. zpr., 2014.
- [ŠO2] I. Punčochář, J. Škach, M. Šimandl a L. Král, “Nonlinear analysis of position estimate in global navigation satellite systems”, *Journal of Physics: Conference Series*, sv. 570, č. 1, s. 1–12, 2014.
- [ŠO3] J. Škach, “Úloha řízení kyvadla pomocí dynamického programování”, in *SVK FAV Studentská vědecká konference: Sborník rozšířených abstraktů*, Pilsen, Czech Republic: University of West Bohemia, 2014, s. 103–104.
- [ŠO4] M. Šimandl, I. Punčochář a J. Škach, “Positioning using GNSS signals and digital maps”, in *Proceedings of the Intelligent Signal Processing Conference 2013 (ISP 2013)*, IET, London, United Kingdom, 2013.
- [ŠO5] J. Škach, *Stanovení polohy pohybujícího se objektu pomocí satelitního systému*, Diplomová práce, Pilsen, Czech Republic: University of West Bohemia, 2013.
- [ŠO6] —, “Lokalizace objektu s využitím mapy dopravní sítě”, in *SVK FAV Studentská vědecká konference: Sborník rozšířených abstraktů*, Pilsen, Czech Republic: University of West Bohemia, 2013, s. 85–86.