

UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
ODDELEK ZA FIZIKO
Magistrski študijski program 2. stopnje Fizika
Matematična fizika

Krištof Špenko

**Uporaba metod strojnega učenja za hitro
ekstrapolacijo sledi delcev v detektorjih na
Velikem hadronskem trkalniku**

Magistrsko delo

MENTOR: prof. dr. Borut Paul Kerševan
SOMENTOR: izr. prof. dr. Rok Pestotnik

Ljubljana, 2022

Zahvala

Rad bi se zahvalil mentorju prof. dr. Borutu Paulu Kerševanu za njegovo budno oko in spremstvo. Prav tako bi se rad zahvalil tudi strokovnemu mentorju Andriiju Usachu za njegovo pomoč pri implementaciji programa na CERN-u. Zahvalil bi se rad tudi somentorju dr. Roku Pestotniku za njegovo pomoč pri pregledovanju magistrskega dela. Hvaležen sem tudi doc. dr. Miha Mihoviloviču za vse nasvete. Na koncu pa bi se rad zahvalil še moji Tei in mojim najbližjim za vso pomoč in podporo, ki so mi jo nudili v času, ko sem pripravljal magistrsko delo. Hvala!

It takes a thousand men to invent a telegraph, or a steam engine, or a phonograph, or a telephone or any other important thing — and the last man gets the credit and we forget the others. He added his little mite — that is all he did. These object lessons should teach us that ninety-nine parts of all things that proceed from the intellect are plagiarisms, pure and simple; and the lesson ought to make us modest. But nothing can do that.

Mark Twain

Uporaba metod strojnega učenja za hitro ekstrapolacijo sledi delcev v detektorjih na Velikem hadronskem trkalniku

IZVLEČEK

V obdobju “Run 3” delovanja Velikega hadronskega trkalnika LHC je zajem podatkov za iskanje procesov nove fizike velikanski. Za zadovoljivo natančne meritve je potrebna zelo kompleksna računalniška rekonstrukcija dogodkov trkov v detektorjih, kjer enega največjih izzivov predstavlja rekonstrukcija sledi nabitih delcev. Poleg natančnosti postopkov rekonstrukcije sledi je bistvena tudi hitrost algoritmov, saj le-ta omejuje količino podatkov, ki jo lahko zajamemo. Obetavno nadgradnjo obstoječih postopkov rekonstrukcije sledi, ki temelji na funkcijskih algoritmih, predstavlja uporaba metod strojnega učenja ter implementacija novih metod na najmodernejših računalniških arhitekturah (GPU - grafične kartice). Za razvoj tovrstnih algoritmov je nujno potrebno razumevanje fizikalnega ozadja, od interakcije delcev s snovjo v detektorju do razumevanja same kinematike iskanih fizikalnih procesov, kot jih napoveduje Standardni model in različni modeli nove fizike. Uspeh postopkov in dosegljive izboljšave je potrebno tudi preveriti na obstoječih podatkih, zajetih na LHC. Cilj magistrskega dela je iskanje in razvoj alternativnih ekstrapolatorskih metod za rekonstrukcijo trajektorij nabitih delcev v magnetnem polju. V ta namen predstavimo tri metode, ki temeljijo na funkcionalni napovedi trajektorij, s čimer se izognemo počasni adaptivni numerični integraciji in omogočimo uporabo zasnovanih algoritmov na najnovejših računalniških arhitekturah. Prva metoda, t.i. parametrizirani ekstrapolator temelji na perturbativnem razvoju popravkov okoli idealnih trajektorij, ki izvirajo točno iz izhodišča. Druga metoda SVD je več-dimenzionalni funkcionalni nastavek, kjer prilagajanje prostih parametrov temelji na zelo osnovnem pristopu učenja na velikem učnem naboru trajektorij preko minimizacije χ^2 . Kot zadnjo metodo predstavimo pravo metodo strojnega učenja metodo GPR, pri kateri na podlagi majhnega nabora učnih trajektorij modeliramo korelacije med meritvami in preko njih opišemo verjetnostno porazdelitev napovedi. Na koncu magistrske naloge vse tri metode testiramo in preverimo, da so njihovi izračuni napovedi dovolj natančni za zanesljivo rekonstrukcijo sledi delcev.

Ključne besede: strojno učenje, regresija z Gaussovimi procesi, osnovni delci, ekstrapolacija, regresija, magnetno polje, detektorji delcev, rekonstrukcija, sledi delcev, dipol

Application of machine learning methods for fast extrapolation of particle tracks in detectors at the Large hadron collider

ABSTRACT

During the “Run 3” period of operation of the Large Hadron Collider LHC, the acquisition of data is gigantic. For accurate measurements, a complex computer reconstruction of the collision events is required, where the reconstruction of particle tracks is one of the biggest challenges. In addition to the accuracy of the track reconstruction procedures, the speed of the algorithms is also essential, as this limits the amount of data that can be captured. The use of machine learning methods and the implementation of new methods on modern computer architectures (GPU - graphical processing unit) show great promise in upgrading existing track reconstruction procedures. The development of such algorithms requires an understanding of the physical background of the problem, from the interaction of particles with matter in the detector to the understanding of the kinematics of the physical processes, as predicted by the Standard Model and possible new physics processes. The success of the procedures must also be checked based on the existing data collected at the LHC. The goal of the master thesis is to find and develop alternative extrapolation methods for the accurate reconstruction of charged particle tracks in the magnetic field. For this purpose, we present three methods based on functional predictions of state vectors, with which we avoid the slow adaptive numerical integration. The first method is called the parametrized extrapolator and it works as a perturbative expansion around ideal tracks, which come exactly from the origin. The second method is based on a multi-dimensional functional ansatz whose parameters we fit by minimizing the χ^2 on a large dataset by using SVD. The last is the GPR method, which fits correlations between sparse measurements, by which it then constructs a probability distribution around the most probable outcome. In the end, we implement and test all three methods and check that they are able to replicate the required accuracy.

Keywords: machine learning, Gaussian process regression, elementary particles, extrapolation, regression, magnetic field, particle detectors, reconstruction, particle tracks, dipole

Kazalo

Seznam slik	11
Seznam tabel	13
Seznam kratic	15
1 Uvod	17
2 Detektor LHCb	21
2.1 LHC	21
2.2 Detektorske postaje detektorja LHCb	23
2.2.1 Iskalec vozlišč	25
2.2.2 Detektorji sevanja Čerenkova	25
2.2.3 Sledilni sistem v LHCb	26
2.2.4 Kalorimetri v LHCb	27
2.2.5 Mionski spektrometer na LHCb	28
2.2.6 Zajemanje podatkov na LHCb	28
2.3 Magnetni dipol na LHCb	30
3 Rekonstrukcija trajektorij nabitih delcev	35
3.1 Prepoznavanje vzorcev	36
3.2 Kalmanov filter	39
3.2.1 Napoved Kalmanovega filtra	40
3.2.2 Filtriranje	41
3.3 Runge-Kutta ekstrapolator	42
4 Simuliranje eksperimenta	47
4.1 Pregled simulacije	48
4.1.1 Generiranje dogodkov	50
4.1.2 Simuliranje odziva detektorjev	52
4.2 Podatkovni seti za razvoj algoritmov	53
4.2.1 Šopkasti podatkovni set	54
4.2.2 Podatkovni set iz simulacij	55
4.2.3 Kvazi-naključno vzorčenje	55
5 Nove metode za hitro ekstrapolacijo trajektorij	59
5.1 Parametrizirani ekstrapolator	59
5.1.1 Vzratna ekstrapolacija	64
5.2 Prilagajanje z metodo SVD	66
5.3 Regresija z Gaussovimi procesi	70

5.3.1	Matematično ozadje	72
5.3.2	Implementacija algoritma	73
5.4	Rezultati	77
5.4.1	Ekstrapolacije naprej	77
5.4.2	Ekstrapolacije nazaj	79
6	Zaključek	83
	Literatura	85
	Dodatek A Gibanje nabitih delcev v magnetnem polju	89
	Dodatek B Izpeljava Kalmanovega filtra	91
	Dodatek C Izguba energije in sipanje	93
	Dodatek D Dodatni testi parametriziranega ekstrapolatorja	97

Seznam slik

1.1	Razpad B_s^0 mezona.	17
1.2	Hadroni odkriti na LHCb.	18
1.3	Feynmanov diagram FCNC.	19
2.1	CERN-ova veriga pospeševalnikov.	23
2.2	Razporeditev detektorskih postaj znotraj eksperimenta LHCb.	24
2.3	Iskalec vozlišč VELO.	25
2.4	Prerez valjaste oblike detektorja VELO.	25
2.5	Detektorja sevanja Čerenkova RICH.	26
2.6	Sledilni sistem.	27
2.7	Kalorimetri.	28
2.8	Mionski spektrometer.	28
2.9	Organizacija HLT2.	29
2.10	Delež RK ekstrapolatorja v rekonstrukcijski programski opremi.	30
2.11	Prerez detektorja LHCb.	30
2.12	LHCb dipolni magnet.	31
2.13	Vizualizacija gostote magnetnega polja.	32
2.14	Vizualizacija divergence in absolutne vrednosti rotorja gostote magnetnega polja.	33
2.15	Izsek animacije odvisnosti z komponente rotorja gostote magnetnega polja.	34
3.1	Definicija pet-dimenzionalnega vektorja stanja.	35
3.2	Tipične karakteristične oblike trajektorij.	37
3.3	Shematična ilustracija metode p -kick.	39
3.4	Shematična ilustracija delovanja Kalmanovega filtra.	41
3.5	Ilustrativna vizualizacija delovanja klasične RK metode.	44
3.6	Študija adaptivnega koraka RK metode.	45
4.1	Primer uporabe Feynmanovih pravil.	49
4.2	Pričakovani sipalni preseki za protonske trkalnike.	50
4.3	Sipalni preseki za nekatere procese v odvisnosti od invariantne mase trka.	50
4.4	Shema generiranja dogodka.	51
4.5	Partonske gostote PDF.	51
4.6	Primeri Feynmanovih diagramov tipičnih za tvorbo $b\bar{b}$ parov.	52
4.7	Korelacija polarnih kotov smeri trajektorije hadronov B	52
4.8	Tri-dimenzionalna vizualizacija simulacije najbolj tipičnih trajektorij.	53
4.9	Vizualizacija simulacije odziva detektorja v programu Geant	54
4.10	Vizualizacija šopkastega podatkovnega seta.	55

4.11	Vizualizacija podatkovnega seta pridobljenega s simulacijami z minimalno pristranskostjo (ang. <i>minimum bias</i>).	56
4.12	Vizualizacija podatkovnega seta z minimalno pristranskostjo pri različnih številih trajektorij in pri različnem načinu vzorčenja teh trajektorij.	56
5.1	Ilustracija začetne ukrivljenosti idealnih trajektorij delcev.	60
5.2	Ilustracija prilagajanja parametriziranega ekstrapolatorja.	62
5.3	Študija natančnosti parametriziranega ekstrapolatorja	64
5.4	Študija natančnosti parametriziranega ekstrapolatorja v odvisnosti od ukrivljenosti trajektorije	65
5.5	Natančnost parametriziranega ekstrapolatorja z Newton-Raphson iterativno shemo.	66
5.6	Konvergenca hiperparametrov SVD ekstrapolatorja.	68
5.7	Primerjava navadnih polinomskih členov s Čebiševimi polinomi.	69
5.8	Vrednosti parametrov SVD ekstrapolatorja.	70
5.9	Študija klasičnih metod strojnega učenja za vzvratno ekstrapolacijo.	71
5.10	Racionalna kvadratna jedrna funkcija v metodi GPR.	75
5.11	Konvergenca natančnosti metode GPR.	76
5.12	Natančnost parametriziranega ekstrapolatorja pri esktrapolacijah naprej skozi magnet.	78
5.13	Študija napovedi Jacobijeve matrike parametriziranega ekstrapolatorja pri esktrapolacijah naprej skozi magnet.	79
5.14	Natančnost SVD in GPR ekstrapolatorja pri esktrapolacijah nazaj skozi magnet.	80
5.15	Študija napovedi Jacobijeve matrike SVD in GPR ekstrapolatorja pri esktrapolacijah nazaj skozi magnet.	81
C.1	Graf moči ustavljanja $-dE/dx$	94
D.1	Študija natančnosti parametriziranega ekstrapolatorja pri ekstrapolacijah med VELO in UT	97
D.2	Študija natančnosti Jacobijeve matrike parametriziranega ekstrapolatorja pri ekstrapolacijah med VELO in UT	97
D.3	Študija natančnosti parametriziranega ekstrapolatorja pri ekstrapolacijah med UT in VELO	98
D.4	Študija natančnosti Jacobijeve matrike parametriziranega ekstrapolatorja pri ekstrapolacijah med UT in VELO	98

Seznam tabel

3.1	<i>Butcher</i> -jeva tabela koeficientov za numerično shemo <i>Cash-Karp</i>	44
4.1	Prispevki različnih procesov ki v simulaciji dogodkov s programom <i>Pythia</i> prispevajo k tvorbi hadronov <i>B</i> znotraj acceptance detektorja <i>LHCb</i> [2].	52
5.1	Časovno testiranje metod za ekstrapolacije naprej skozi magnet . . .	63
5.2	Časovno testiranje ekstrapolatorjev v vzvratno smer skozi magnet. . .	77

Seznam kratic

Kratice

ALICE Veliki eksperiment ionskih trkov

BAU barionska asimetrija vesolja

CERN Evropska organizacija za jedrsko znanost

DELPHI Detektor z identifikacijo leptonov, fotonov in hadronov

FCNC okus-spreminjajoči nevtralni tok

GPR regresija z Gaussovimi procesi

GPU grafična kartica

HLT Visoko nivojski prožilec

L0 začetni nivo HLT prožilca

LHC Veliki hadronski trkalnik

LHCb LHC eksperiment *b*-kvarka

LHCf LHC sprednji eksperiment

Linac 2 Linearni pospeševalnik 2

MAP maximum a posteriori

MC Monte Carlo

mip minimalno ionizirani delec

NR Newton-Raphson

PBS Protonski pospeševalni sinhrotron

PS Protonski sinhrotron

QCD kvantna kromodinamika

RBF radialne bazne funkcije

RICH detektor sevanja Čerenkova

RK Runge-Kutta

SM Standardni model

SPS Superprotonski sinhrotron

SVD razcep singularnih vrednosti

TOTEM Meritev skupnega elastičnega in difrakcijskega sipalnega preseka

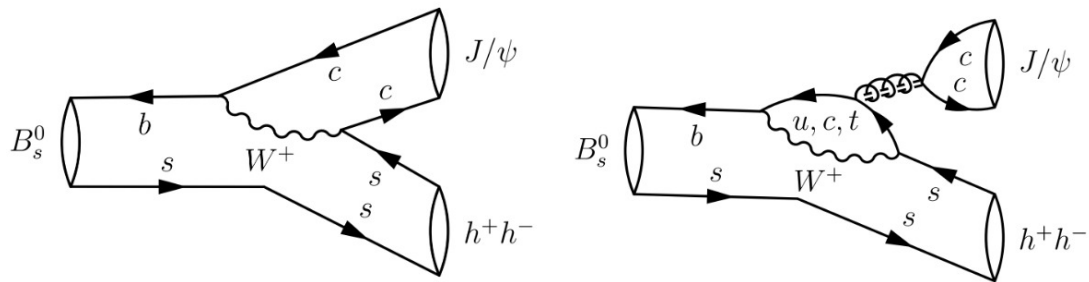
UT Detektor vzpenjajočih trajektorij

VELO Iskalec vozlišč (ang. *vertex locator*)

Poglavje 1

Uvod

Zgodovinsko dejstvo je, da ljudi zanima svet okoli njih. Tako so že antični Grki razglabljali o sestavi snovi. Anaxagoras je predpostavljal, da se snov spreminja zaradi premikanja nevidnih delcev, ki jih je Demokrit poimenoval atome. Počakati smo morali do l. 1897, da se je začela moderna veda o osnovnih delcih, ko je J. J. Thompson odkril elektron. Ernest Rutherford je nato preko sipanja elektrona potrdil obstoj jedra atoma. Antidelce je nato l. 1932 odkril Carl Anderson z meritvijo pozitrona iz kozmičnih žarkov. Znanje o našem vesolju smo tako opisali v teorijah splošne relativnosti, s katero opišemo gravitacijo, in Standardnega modela (SM)[3], s katerim opišemo preostale tri osnovne sile; elektromagnetno, šibko in močno silo. Z eksperimenti skušamo preveriti veljavnost naših teoretičnih modelov in čeprav jih vedno znova in znova potrdimo s čedalje natančnejšimi meritvami, vemo da nimamo popolne prave teorije, saj npr. ne moremo združiti teh obeh pristopov. Vendar pa to ni edini razlog, zaradi katerega dvomimo v SM. Če se ozremo po našem vesolju opazimo, da je skoraj vso vesolje sestavljeno iz snovi in zelo malo antisnovi[4]. Ta asimetrija je eden izmed izzivov sodobne fizike, ki ga želimo razumeti. Nujen pogoj za barionsko asimetrijo vesolja (BAU)[4] je kršitev CP simetrije. Teoretični operaciji C in P skupaj zamenjata vse delce v antidelce in obrneta vse smeri koordinatnega sistema. Če fizika delcev ni invariantna pod to skupno CP operacijo, potem se snov obnaša drugače od antisnovi. Študija kršitve CP simetrije je ena izmed glavnih motivacij za delovanje eksperimenta LHCb. Trenutni izmerjeni nivo CP kršitve je premajhen, da bi popolnoma opisal asimetrijo antisnovi. Nova fizika bi lahko predstavila še neodkrite delce in oz. posrednike osnovnih sil, ki bi lahko opisali ta fenomen.

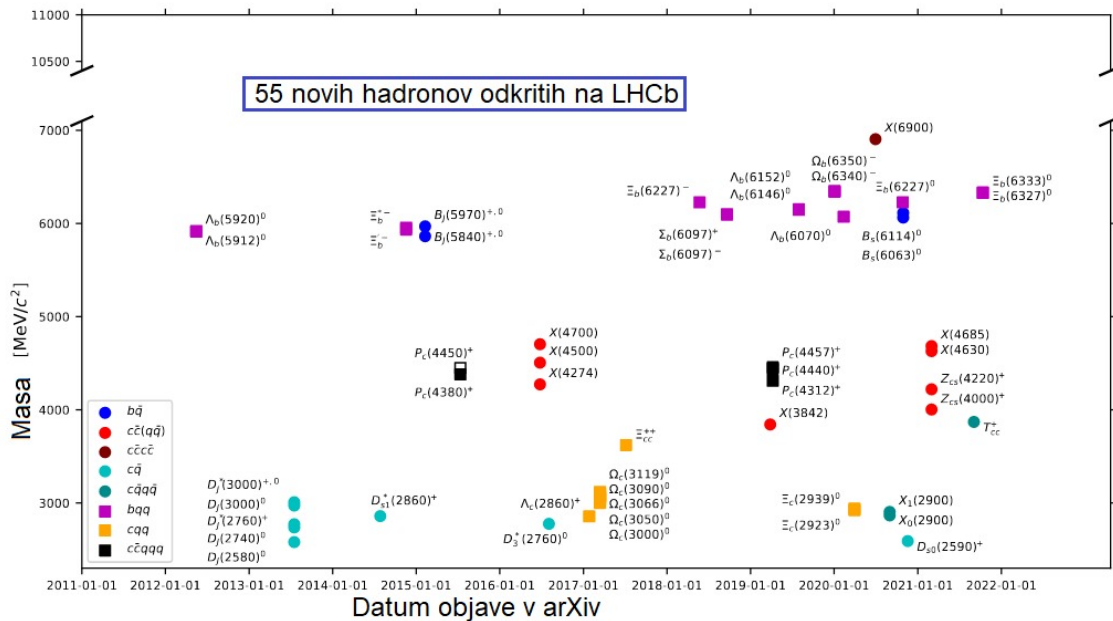


Slika 1.1: Primer dveh Feynmanovih diagramov, ki opisujejo razpad mezona B_s^0 . Na levi vidimo drevesni diagram najnižjega reda, na desni pa t.i. pingvinski diagram. Oznaka h lahko označuje pion π ali kaon K [5].

Ena izmed potencialnih vej iskanja nove fizike temelji na analizi t.i. fizike B , ki preučuje procese hadronov, ki vsebujejo kvarke b . Prav za te namene se specializira eksperiment LHCb in ker deluje na hadronskem trkalniku, imamo priložnost študirati nove hadrone kot so B_s^0 in B_c^+ mezoni ter hadrone b z izjemno natančnostjo. Posebno zanimiv je razpad $B_s^0 \rightarrow J/\psi\phi$, saj omogoča natančno merjenje nivoja CP kršitve v SM[6]. Dva Feynmanova diagrama razpada sta prikazana na sliki 1.1. Tipično tovrstni procesi potekajo eden za drugim v dolgi in izjemno prepleteni verigi procesov, ki vključujejo sipanja, anihilacijo in razpad delcev. Tako recimo J/ψ mezon najprej razpade na dva leptona. Izbran teoretični model lahko potrjujemo ali zavračamo tudi na podlagi verjetnosti, kolikokrat opazimo določene procese, preko t.i. razvejitvenih razmerij (ang. *branching ratio*). Ta znaša za razpad J/ψ mezona na dva leptona[7]:

$$\text{BR}(J/\psi \rightarrow l^+l^-) = (11.932 \pm 0.032)\%, \quad (1.1)$$

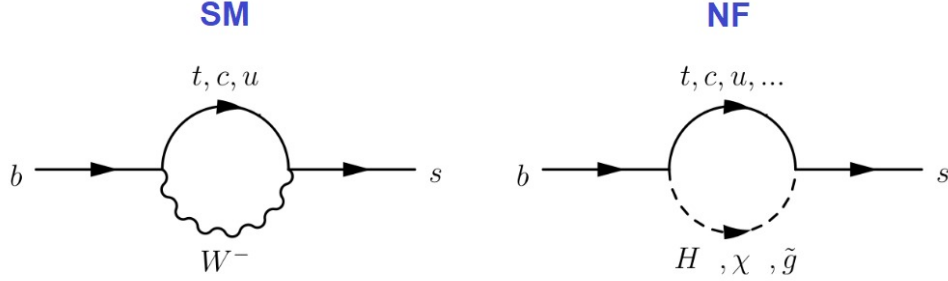
kjer sta leptona lahko iz katere koli generacije; elektron, mion ali tauon in njihov pripadajoči antidelec[7]. Tukaj SM napove, da je verjetnost za razpad na katerokoli izmed treh generacij odvisna zgolj od mase leptonov in ne od njihovega okusa. Ta pojav imenujemo *leptonska univerzalnost* in jo prav tako želimo vedno znova bolj natančno izmeriti z eksperimenti. V primeru, da bi z eksperimenti opazili drugačne frekvence takšnih procesov, bi to lahko nakazovalo na novo fiziko. Eksperiment LHCb je od l. 2011 do sedaj odkril 55 novih hadronov, ki jih lahko vidimo na sliki 1.2.



Slika 1.2: Graf prikazuje 55 novih hadronov odkritih na eksperimentu LHCb od l. 2011 (do konca 2021). Sedem dodatnih hadronov je bilo odkritih na eksperimentih CMS in ATLAS, ki pa niso prikazani tukaj[8].

Vsak izmed njih ima tudi številne razpadne načine, ki jih želimo eksperimentalno zaznati. Med številnimi razpadi hadronov b obstajajo tudi zelo redki še ne izmerjeni procesi, ki vsebujejo okus-spreminjajoče nevtralne tokove (FCNC - ang. *Flavour-Changing Neutral-Currents*) pri katerih fermioni spremenijo okus ne da bi zamenjali predznak električnega naboja. V SM so te procesi prepovedani na nivoju drevesnih

Feynmanovih diagramov in so dovoljeni šele ob prisotnosti zank. Zato so meritve teh procesov izjemno občutljive na morebitno novo fiziko onkraj SM, kjer bi novi delci posredno lahko vplivali na dinamiko prehodov, kot jih vidimo prikazane na sliki 1.3.



Slika 1.3: Feynmanov diagram FCNC prehoda $b \rightarrow s\gamma$ v SM (levo) in s prispevki delcev nove fizike onkraj SM (desno)[9].

Eksperiment LHCb tako skuša rekonstruirati celotne razpadne verige na podlagi končnih stanj delcev, ki jih zaznamo z detektorskimi postajami. Rekonstrukcija poteka v obratno smer razpadne verige, pri čimer razpadne produkte združujemo na podlagi kinematike rekonstruiranih delcev. Z detektorji zaznamo pot prehoda končnih delcev, izmerimo njihove gibalne količine in energije ter jih identificiramo. Iz meritev rekonstruiramo njihovo trajektorijo in ocenimo vozlišča razpadnih procesov. Preko ocenjenih četvercev gibalnih količin lahko nato rekonstruiramo invariantne mase razpadlih delcev. V primeru dvo-delčnega razpada jo izračunamo po formuli:

$$\begin{aligned}
 M^2 &= (p_{1\mu} + p_{2\mu})(p_1^\mu + p_2^\mu) \\
 &= (E_1 + E_2)^2 - \|\vec{p}_1 + \vec{p}_2\|^2 \\
 &= m_1^2 + m_2^2 + 2(E_1 E_2 - \vec{p}_1 \cdot \vec{p}_2).
 \end{aligned}
 \tag{1.2}$$

Za natančen izračun potrebujemo čim bolj izmerjene gibalne količine in energije delcev. Za zadovoljivo natančne meritve teh količin je potrebna zelo kompleksna računalniška rekonstrukcija dogodkov trkov v detektorjih, kjer je eden izmed najbolj zahtevnih izračunov rekonstrukcija sledi delcev. Pri fizikalni analizi redkih procesov, ki bi morda pokazali dokaze za fiziko onkraj SM, pa je potrebna rekonstrukcija velike množice procesov. Hitrost rekonstrukcijskih algoritmov posledično omejuje količino podatkov, ki jih zajamemo in analiziramo. Obetavno nadgradnjo standardnih obstoječih postopkov rekonstrukcije sledi delcev, ki temelji na iterativni numerični integraciji enačb gibanj za nabite delce v magnetnem polju, predstavlja uporaba metod strojnega učenja ter implementacije novih metod na najmodernejših računalniških arhitekturah.

V naslednjem poglavju bomo predstavili obliko in delovanje detektorja LHCb, ki vključuje pregled nad detektorskimi postajami in postopkom zajemanja podatkov. Na koncu drugega poglavja se bomo poglobili v analizo magnetnega polja, ki ga ustvarja dipolni magnet LHCb. Tretje poglavje vsebuje opis postopka rekonstrukcijske programske opreme s poudarkom na ekstrapolacijah trajektorij z Runge-Kutta numerično integracijo enačb gibanj za nabite delce v magnetnem polju. V četrtem poglavju je opisana simulacija eksperimenta skupaj s podatkovnimi seti, ki smo jih uporabili za učenje novih metod. V petem poglavju so predstavljene tri nove ekstrapolatorske metode. Parametrizirani ekstrapolator temelji na dvo-dimenzionalni

parametrizaciji koeficientov prilagoditvene funkcije razvite okoli idealnih trajektorij, ki izvirajo iz izhodišča. Uporaba SVD dekompozicije omogoča prilagajanje kompleksnega več-dimenzionalnega nastavka na podlagi minimizacije χ^2 za predoločen sistem meritev iz učnega podatkovnega seta. Kot zadnja metoda je predstavljena regresija z Gaussovimi procesi (GPR), ki modelira napoved na podlagi korelacij med majhnim številom meritev. Čisto na koncu so predstavljeni rezultati testiranja metod na rekonstrukcijski programski opremi LHCb.

Poglavje 2

Detektor LHCb

Evropska organizacija za jedrske raziskave (CERN)[10] je laboratorij namenjen za preučevanje fizike pri visokih energijah. Nahaja se na meji med Francijo in Švico blizu Ženeve. CERN je leta 1954 ustanovilo dvanajst držav, sedaj pa ima 23 članic in še številčnejšo izven člansko kolaboracijo. Primarni cilj je raziskovati osnovne delce in interakcije med njimi z uporabo tehnološko napredne infrastrukture Velikega hadronskega trkalnika LHC za poganjanje eksperimentov. LHCb je eden izmed eksperimentov, ki merijo procese, ki nastanejo ob trkih protonov v LHC, ki se specializira za raziskave fizike okusov. V to spada preučevanje kršitve CP simetrije¹ in zaznavanja redkih razpadov hadronov, ki vsebujejo kvarke b , preko česar iščemo procese nove fizike.

2.1 LHC

The Large Hadron Collider (LHC)[11] ali Veliki hadronski trkalnik je bil zgrajen za preučevanje fundamentalnih interakcij med osnovnimi delci. Te fizikalne procese lahko zaznamo pri trkih visokoenergijskih delcev, pri katerih lahko pride do pretvorbe energije v maso in tako do nastanka novih delcev². V osnovi trkalnik pospešuje dva curka protonov in jih nato trči skupaj s težiščno energijo tudi do 13 TeV. LHC je zadnji del CERN-ove verige pospeševalnikov in obratuje že od leta 2010. Ima obseg 26,7 km in se nahaja 170 m pod zemljo, na meji med Francijo in Švico.

Pospeševanje protonov poteka na naslednji način: najprej se z ionizacijo vodikovih atomov izvleče protone, ki se jih nato pospeši z linearnim pospeševalnikom Linac 2 (50 MeV). Ti so nato predani PBS - protonoskemu pospeševalnemu sinhrotronu (1,4 GeV), v katerem se protoni razdelijo v 4 curke za maksimalni izkoristek. Curki se v tem delu tudi zberejo. Po združitvi pa žarek preide v PS - protonski sinhrotron (26 GeV), v katerem se hitrost protonov približa svetlobni hitrosti, zato se dodatno pospeševanje opazi kot večanje relativistične mase. Curek je nato poslan v SPS - superprotonski sinhrotron, v katerem doseže skupno energijo 450 GeV. Žarek je nato razdeljen na dva dela in predan LHC[10].

¹ C simetrija je izraz, s katerim opišemo pojave, ki se ne spremenijo, ko zamenjamo delce z njihovimi antidelci. P simetrija pa pomeni, da se pojav ne spremeni, če obrnemo koordinatni sistem, torej zamenjamo levo in desno stran[6].

² $E = mc^2$ Einsteinova formula.

LHC je razdeljen na oktante v obliki lokov. Sestavljen je iz dveh vakuumskih cevi, po katerih dva žarka protonov usklajeno potujeta v nasprotnih si smereh. LHC vsebuje več različnih komponent. Radiofrekvenčne resonančne votline so namenjene pospeševanju protonov. Ker pa ne pospešujejo celega žarka enakomerno, se ta preoblikuje v curek 2835 gruč (ang *bunch*) protonov. V vsakem skupku je 115.000 milijonov protonov. Vsak skupek je dolg okoli 7,55 cm. Za usmerjanje curka skrbi 1232 superprevodnih dipolnih magnetov z jakostjo magnetnega polja do 8,33 T. Magneti so ključnega pomena za nadzor curka, njihova zmogljivost pa predstavlja omejitve pospeševalnika. Curek kolimiramo s 392 kvadrupolnimi magneti, ki zožajo curek na širino $16,7 \mu\text{m}$, kar je približno enako debelini človeških las. Višji multipolni magneti skrbijo za nelinearne popravke žarka. Curka lahko trčita v 4 interakcijskih točkah, v katerih se nahajajo različni detektorji. Protoni so ob končanem pospeševanju tako hitri, da v eni sekundi naredijo 11 000 obhodov. Ko curka združimo, gruč protonov naletavajo s frekvenco 40,06 MHz, kar traja nekaj ur. Ob vsakem naletu dveh skupkov protonov dobimo okoli 25 *dogodkov*³[11].

V LHC lahko z visokoenergijskim curkom protonov dosežemo visoke frekvence trkov, pri katerih prihaja do fizikalnih procesov oz. dogodkov. Frekvenca, s katero nastajajo dogodki specifičnega fizikalnega procesa, je podana z $R = \frac{dN}{dt} = L\sigma$, kjer je L *svetilnost* žarka, σ pa predstavlja *sipalni presek* za obravnavani fizikalni proces. Sipalni presek je popolnoma odvisen od samega fizikalnega pojava, svetilnost pa od pospeševalnikove zmogljivosti. Svetilnost žarka pospeševalnika ocenimo z izrazom v enačbi 2.1, kjer je n_b število skupkov, N_b število delcev v skupku, f_{rev} frekvenca kroženja, σ_t transverzalna velikost curka, σ_z je dolžina skupka, F je geometrijski popravek, ki je posledica kota križanja curkov θ_c .

$$L = \frac{n_b N_b^2 f_{rev}}{4\pi\sigma_t^2} F \quad (2.1)$$

$$F = \left(1 + \left(\frac{\theta_c \sigma_z^2}{2\sigma_t}\right)^2\right)^{-1/2} \quad (2.2)$$

LHC omogoča nalete na 4 različne točke, v katerih lahko trčimo curka. Kot prikazano na sliki 2.1, se na interakcijskih točkah nahajajo različni detektorji[12]:

- ATLAS - *A Toroidal LHC Apparatus* oz. Toroidna naprava LHC služi kot splošni detektor.
- CMS - *Compact Muon Solenoid* oz. Kompaktni mionski solenoid je prav tako detektor za splošno uporabo.
- LHCb je detektor namenjen natančnim meritvam kršitev simetrije CP ⁴ in detekciji razpadov hadronov B .
- ALICE - *A Large Ion Collider Experiment* oz. Veliki eksperiment ionskih trkov služi kot splošni detektor pri visokoenergijskih trkih težkih ionov, kjer LHC pospešuje težke ione (svinčeva jedra). Namenjen je predvsem opazovanju močne interakcije Standardnega modela.

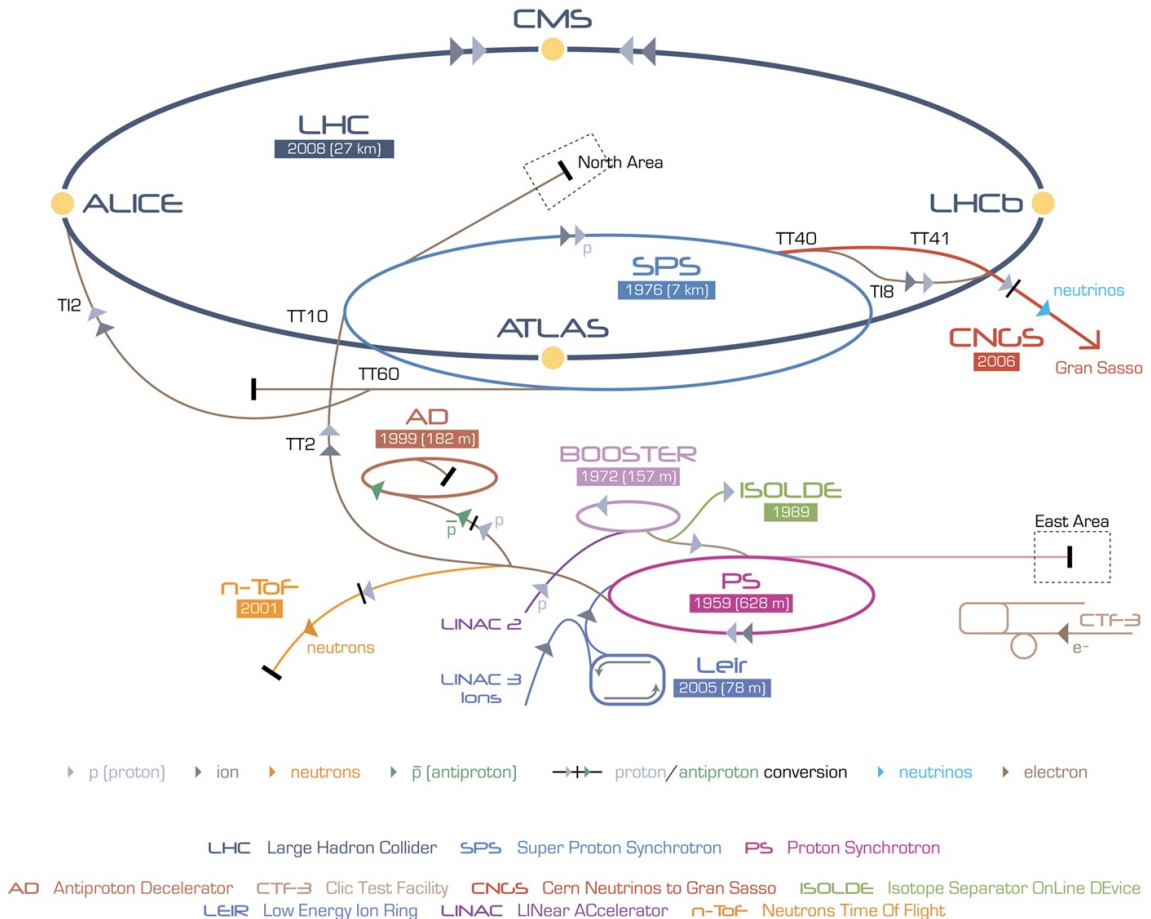
³Dogodek (ang *event*) - klasično gledano je to trk, gre pa za dobro lokaliziran pojav, pri katerem pride do fundamentalne interakcije med osnovnimi delci.

⁴Simetrija CP zahteva, da zakoni fizike ostanejo enaki, če prezrcalimo prostor in zamenjamo vse delce z njihovimi antidelci.

2.2. Detektorske postaje detektorja LHCb

Poleg teh pa sta blizu točk, v katerih lahko trčimo curka, postavljena še dva sklopa detektorjev:

- LHCf je namenjen merjenju nevtralnih delcev oddanih v smeri curka protonov.
- TOTEM je namenjen meritvam celotnega sipalnega preseka in preučevanju sipanja delcev.



Slika 2.1: CERN-ova veriga pospeševalnikov, skupaj z eksperimenti.[10]

2.2 Detektorske postaje detektorja LHCb

Eksperiment LHCb poskuša čim bolj natančno izmeriti razpade mezonov B ter drugih hadronov, ki vsebujejo kvark b , do katerih prihaja na LHC v večji količini kot pri manjših trkalnikih. Razpadi določenih mezonov B so pomembni pri opazovanju kršitve CP simetrije, spet drugi redkejši razpadi pa lahko kažejo sledove nove fizike. Pri opazovanju teh specifičnih razpadov je ključno, da jih razberemo iz množice procesov ozadja. Pomembno je razlikovanje med primarnim vozliščem trka protonov in sekundarnim vozliščem razpada mezona B ⁵. Ko pride do naleta dveh skupkov protonov, pride ob trku dveh protonov do interakcije med njihovimi gradniki *partoni*⁶.

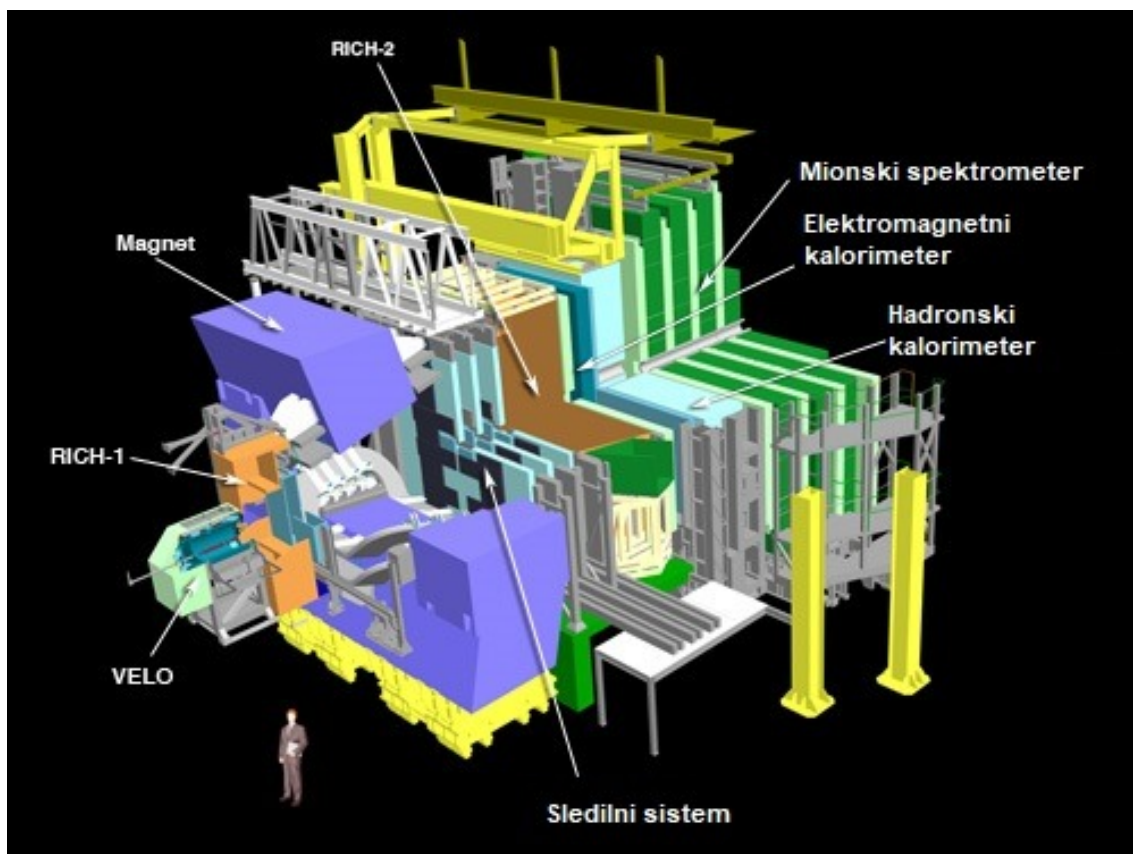
⁵vozlišče (ang. *vertex*), izraz se uporablja za lokacijo fizikalnih procesov kot so razpadi ali trki delcev

⁶*Partonski model* (R. Feynman): pri opazovanju interakcij med visokoenergijskimi hadroni lahko zanemarimo relativno gibanje partonov (parton - sestavni del hadrona).

Poglavje 2. Detektor LHCb

Pri tem se lahko kinetična energija delcev pretvori in pride do nastanka masivnejših in bolj eksotičnih osnovnih delcev, ki jih poimenujemo *začetna stanja*. Taki delci niso stabilni in hitro razpadejo v druge bolj obstojne delce, ki jih lahko zaznamo z detektorjem in jih poimenujemo kot *končna stanja*. Na podlagi izmerjenih vektorjev četvercev končnih stanj lahko določimo mirovno maso začetnega stanja.

Naloga eksperimenta LHCb je primarno opazovati razpade delcev, ki vsebujejo kvark b oz. njegov antidelec. Celotna konstrukcija tehta 4.500 ton in je oblikovana specifično za izluščenje teh ključnih delcev mezonov B in produktov njihovih razpadov. Mezioni B nastanejo ob trkih protonskih žarkov in se tipično gibljejo blizu žarkovne osi. Medtem ko preostali LHC eskperimenti, kot npr. ATLAS, obdajajo točko trka žarkov iz vseh smeri v obliki čebule, ima eksperiment LHCb razporejene detektorje v eno smer, pri čimer so plasti detektorjev razporejene kot knjige na polici, ena za drugo, do dolžine 20 m, kot prikazuje shema na sliki 2.2[1].

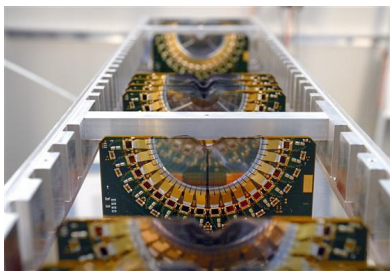


Slika 2.2: Razporeditev detektorskih postaj znotraj eksperimenta LHCb. LHCb je enostranski detektor, ki lahko zazna delce le v eno smer, ko odletijo po nastanku ob trku protonov. Detektorske postaje so postavljene pravokotno glede na žarkovno os, njihova površina pa se večja bolj kot se oddaljujemo od interakcijske točke trkov. Tako dobivamo meritve v območju stožca, saj se trajektorije delcev tipično širijo v obliki šopka rož[13].

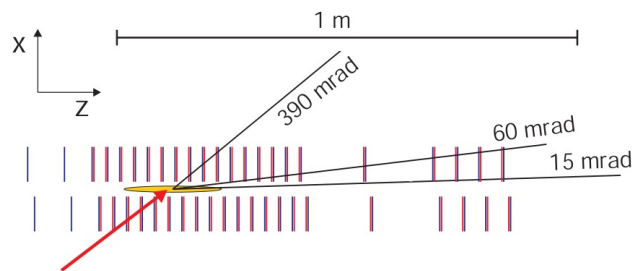
Podobno kot pri ostalih eksperimentih LHC ima vsaka posamezna plast detektorjev svoj poseben namen pri zaznavanju različnih delcev, ki nastanejo pri trkih protonskih žarkov. Skupno lahko tako celoten detektor izlušči informacijo o identiteti, gibalni količini, energiji in poti vsakega novo nastalega delca, z dovolj veliko resolucijo, da lahko razlikuje med milijardami delcev.

2.2.1 Iskalec vozlišč

LHCb je sestavljen iz večjega števila poddetektorjev oz. detektorskih postaj. Večina eksotičnih delcev, ki so zanimivi za odkrivanje nove fizike je zelo kratkoživih in tipično razpadejo še preden bi lahko trčili v katero koli izmed detektorskih plasti. Zato je za rekonstrukcijo izjemno pomembno, da dobimo čim več točnih informacij o začetnem poteku sledi delcev. V neposredni bližini obdaja točko trka protonskih žarkov t.i. iskalec vozlišč (VELO - *vertex locator*). Njegova naloga je izmeriti razdaljo med točko, kjer pride do trka protonskih žarkov, in točko, kjer razpadejo delci kot npr. mezoni B . Ta razdalja tipično znaša okoli 12 mm. Ker gre za relativno veliko razdaljo v primerjavi s preostalimi razpadi, lahko na podlagi te razdalje relativno hitro izluščimo meritve, ki se navezujejo zgolj na razpade mezonov B . Za točno določevanje te razdalje se uporablja 42 gosto naseljenih silicijevih detektorjev (ang. *silicon detectors*). Ker večina mezonov B po nastanku potuje relativno blizu žarkovne osi, ima iskalec vozlišč cilindrično obliko. Ker potrebujemo izjemno resolucijo meritev (stotinko milimetra), so ti detektorji zelo občutljivi. Medtem ko se protonski žarki vcepljajo v žarkovni cevovod, je VELO odmaknjen od žarkovne osi in neaktiven. Šele ko se žarki stabilizirajo, se slicijski elementi mehansko premaknejo v bližino žarkovne osi za merjenje[13].



Slika 2.3: Iskalec vozlišč (VELO - *vertex locator*)[13].



Slika 2.4: Prerez valjaste oblike detektorja VELO - delci, ki nastanejo ob trkih se širijo v obliki stožca, zato zadanejo v vsaj 3 rezine detektorskih slojev[1].

2.2.2 Detektorji sevanja Čerenkova

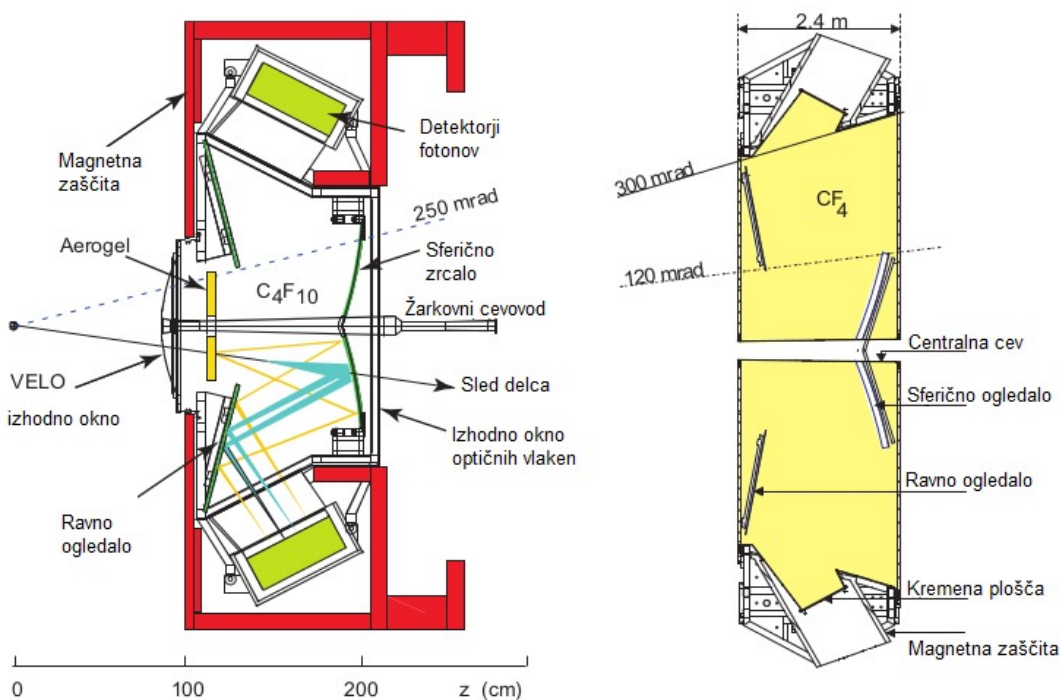
Eksperiment LHCb vsebuje tudi dva detektorja za merjenje sevanja Čerenkova[14] (RICH - *ring imaging Cherenkov detectors*), ki je analogen pojav preboju zvočnega zidu, vendar za delce, ki se gibljejo s hitrostjo večjo od hitrosti svetlobe v danem materialu. Sevanje Čerenkova se tipično širi v obliki stožca (v obratno smer gibanju delca), ki ga detektorji RICH zbirajo skupaj na vrsti senzorjev s pomočjo zrcal. Veliko končnih stanj mezonov B je po obliki topološko zelo podobnih, vendar jih lahko med seboj razlikujemo na podlagi njihovih razpadnih produktov pionov ali kaonov. Detektorji RICH so pomemben vir informacij za identifikacijo razpadnih produktov kot so pioni, kaoni in protoni[13].

Pri pojavu sevanja Čerenkova nabiti delci polarizirajo atome, ko se premikajo čez dielektrično snov, tako da premaknejo elektrone s svojim električnim poljem. Razmaknjeni elektroni se nato vrnejo v svojo ravnovesno lego in pri tem izsevajo fotone. Če se leteči delec giblje s hitrostjo večjo od hitrosti svetlobe v tej snovi, potem pride do konstruktivne interference svetlobe, ki jo izsevajo razmaknjeni elektroni. Tej

svetlobi pravimo sevanje Čerenkova in je izsevana v obliki stožca s karakterističnim kotom, podanim z enačbo 2.3:

$$\cos(\theta_c) = \frac{1}{n} \sqrt{1 + \left(\frac{m}{p}\right)^2} = \frac{1}{n\beta} \quad (2.3)$$

pri čimer je n označen lomni količnik snovi, p je gibalna količina delca, m njegova masa in β je hitrost delca deljena s svetlobno hitrostjo. Preko te zveze lahko identificiramo delec z maso m na podlagi izsevanega stožca Čerenkove svetlobe, če poznamo njegovo gibalno količino. Problem enačbe 2.3 pa je v tem, da pri visokih gibalnih količinah ne more več učinkovito razlikovati med delci. Ta problem lahko do neke mere ublažimo tako, da uporabimo tri različne materiale z različnimi lomnimi količniki za različne režime gibalnih količin[1].

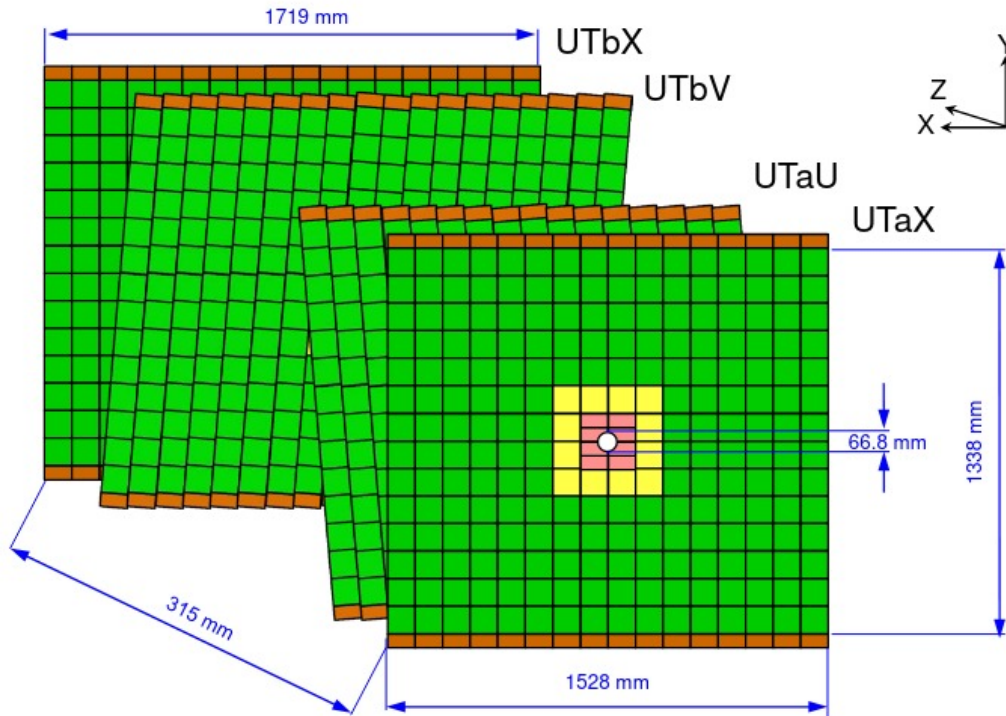


Slika 2.5: Detektorja sevanja Čerenkova, na levi RICH1 in na desni RICH2[1].

2.2.3 Sledilni sistem v LHCb

LHCb vsebuje dva sledilna sistema (ang. *tracking system*)[15] enega pred in enega za magnetom, ki je opisan v poglavju 2.3. Oba sta sestavljena iz štirih zaporednih navpičnih kvadratnih sledilnih plošč, ki so postavljene pod rahlo različnimi koti, ene na drugo kot prikazuje slika 2.5. Taka postavitev omogoča merjenje obeh prostorskih koordinat x in y . V sredini plošče okoli žarkovne osi, so postavljeni zelo občutljivi silicijevi mikro trakovi. Ko nabiti delci preletijo te trakove, oddajo nekaj energije, ki vzbudi elektrone v polprevodniku iz valenčnega v prevodni pas. Ti povzročijo električen tok, ki ga ojačanega zaznamo kot signal s katerega lahko razberemo informacijo o prehodu izvornega delca preleta. Na robovih sledilnih plošč je nastavljenih na tisoče slamnatih cevi polnjenih s plinom[15]. Nabiti delci ob prehodu skozi le-te

zopet oddajo energijo, ki tokrat povzroči ionizacijo plina. Lokacijo o prehodu delca razberemo na podlagi časa, ki ga porabi ionizirani elektron, da doseže anodno žico v sredini cevi. Sledilna sistema sta nastavljena na vsaki strani magneta, kjer se zaradi močnega magnetnega polja sledi delcev najbolj ukrivijo. Na podlagi točnih meritev na vsaki strani magneta, lahko rekonstruiramo ukrivljeno trajektorijo, iz katere ugotovimo gibalno količino delca[16].



Slika 2.6: Nabor štirih detektorskih plošč v postavitvi (x-u-v-x) iz silicijevih mikrotrakov tvori detektor vzpenjajočih se sledi delcev (UT - *upstream tracker*) [16].

2.2.4 Kalorimetri v LHCb

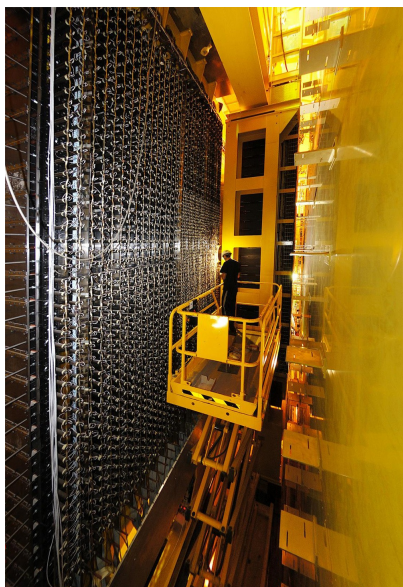
Sistem kalorimetrov je namenjen merjenju, identifikaciji in zaustavljanju hadronov, elektronov in fotonov. Naloga kalorimetra je, da absorbira vso energijo delca in ga ustavi, pri tem pa odda signal, ki je sorazmeren energiji delca. Med prehodom visokoenergijski delci interagirajo s snovjo in tvorijo t.i. sekundarne delce, ki imajo manjšo energijo. Dogajanje se ponovi tudi za sekundarne delce. Tako dobimo iz enega delca plaz sekundarnih delcev⁷. Kalorimetri to s pridom izkoriščajo in so narajeni tako, da delec čimprej tvori prho (pasivni elementi), energija celotne prhe pa se odloži znotraj kalorimetra. Kalorimetra sta sestavljena iz slojev jeklenih oz. svinčenih in scintilatorskih plošč. Če nabiti delec prečka scintilatorsko ploščo, povzroči v snovi razdejanje oblike elektroni-vrzeli. Ko se te rekombinirajo, oddajo UV svetlobo, ki jo zberemo in preko optičnih kablov pošljemo na fotopomnoževalke, od koder s fotodiodami dobimo signal, ki ga lahko nato uporabimo za določitev energije delcev.

⁷Z izrazom prha delcev (ang *particle shower*) opišemo proces, ko iz enega delca nastane plaz oz. kaskada novih delcev.

Elektromagnetni kalorimeter je narejen tako, da ustavi predvsem elektrone. V tem delu detektorja želimo ustaviti vse fotone in elektrone. V hadronskem kalorimetru želimo zaznati samo hadrone in jih v tem delu tudi ustaviti, čemur je prilagojena konstrukcija (čim težji oz. gostejši material). Slika 2.7 prikazuje del hadronskega kalorimetra na LHCb[1].

2.2.5 Mionski spektrometer na LHCb

Mioni so delci, zelo podobni elektronom, vendar z večjo maso. Pogosto nastanejo kot končni produkti razpadov mezonov B , zaradi česar so zelo pomembni za meritve na detektorju LHCb. Mioni so zelo prodorni, zato se ne absorbirajo v detektorju. S spektrometrom, ki se nahaja v obliki ene plasti pred kalorimetri in zadnjih štirih detektorskih plasti, lahko izmerimo čas, ki ga mioni porabijo za prehod. Skupna površina vseh plasti skupaj ima velikost košarkaškega igrišča 435 m^2 . Zadnje plasti so med seboj ločene z debelimi železnimi ščitnimi filtri. Vsaka postaja vsebuje prekate polnjene s kombinacijo treh plinov; ogljikovega dioksida, argona in tetrafluorometana. Mioni ob prehodu zopet interagirajo z mešanico plinov, žične elektrode pa nato zaznajo to interakcijo. Slika 2.8 prikazuje del mionskega spektrometra na LHCb[13].



Slika 2.7: Kalorimetri (ang. *calorimeters*)[13].



Slika 2.8: Mionski spektrometer[13].

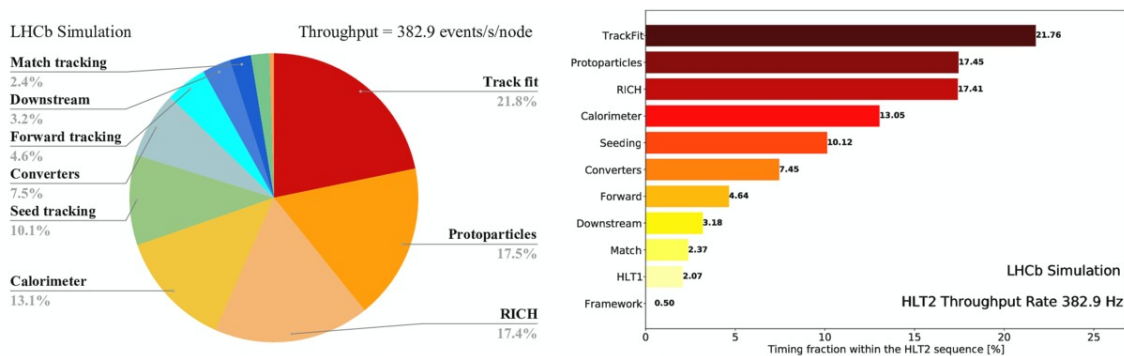
2.2.6 Zajemanje podatkov na LHCb

LHC generira dogodke ob trkih protonov z visoko frekvenco, okoli 1 GHz ($40 \text{ MHz} \times 25$ dogodkov na nalet skupkov). Le okoli en odstotek bo vseboval delce, ki bodo prečkali enoročno postavitev detektorja LHCb, torej bo frekvenca le-teh okoli 10 MHz. Frekvenca dogodkov, ki vsebujejo razpade kvarkov b , pa je zgolj 15kHz. Razpadi mezonov B , ki so zanimivi za analizo pri iskanju nove fizike, pa so samo majhen del teh in znašajo zgolj kakšen Hz. Eksperiment LHCb ima za izbor takšnih dogodkov namenjen prožilni sistem (ang. *Trigger*), ki poskuša s čim boljšo učinkovitostjo

izluščiti dogodke zanimive za razpade B , pri čimer je frekvenca dogodkov, ki jo lahko beležimo omejena z računsko močjo na okoli 2 kHz.

V prvem nivoju prožilnega sistema L0[17] poskušamo preko prepoznavanja vzorcev zmanjšati frekvenco na 1 MHz. Gre za specifično prilagojeno elektroniko, ki s svojo hitro odzivnostjo omogoča uspešno branje relevantnih podatkov in je vgrajena v sam detektor. Tukaj moramo še enkrat poudariti, da večina trkov protonov ne vsebuje zanimive fizike, vsaj v smislu meritev primernih za natančno analizo podatkov. Tipično iščemo delce, ki imajo čim večjo relativno komponentno gibalne količine p_T pravokotne na smer žarkovne osi. Prav tako vsebuje L0 dve posebni plasti znotraj detektorja VELO, s katerima izračuna lokacijo primarnega vozlišča z natančnostjo 3 mm. Skupaj z elektroniko v kalorimetru in mionskem spektrometru pa s tem izračuna zelo grobo rekonstrukcijo trajektorij. Preko te grobe informacije lahko izločimo npr. nezanimive nizkoenergijske dogodke in dogodke, ki so nastali ob trku več kot samo dveh protonov hkrati (v takih primerih je rekonstrukcija namreč močno otežena). Zavrženi dogodki so za vedno izgubljeni.

Temu prožilcu sledita kasneje še dva nivoja, ki ju skupaj imenujemo **HLT** - prožilec visokega nivoja, ki zmanjša frekvenco na 2 kHz. Medtem ko je L0 vgrajena elektronika, je HLT zelo sofisticiran računalniški program, ki se poganja na računalniški gruči. Prvi del algoritmov HLT1 najprej preveri kandidate z visoko p_T z dostopom do še več informacij iz preostalih plasti detektorja, vendar zgolj iz območij, ki jih je preletel posamezen kandidat (predvsem informacija iz VELO in sledilnega sistema). Za zavračanje kandidatov se uporabljajo različni kriteriji glede na obravnavani delec, torej ali gre za trajektorijo elektrona, fotona ali hadronov. Ko je kandidat zavržen, se dogodek izbriše. Drugi nivo HLT2 pa izvede celotno rekonstrukcijo dogodkov z uporabo celotne preostale informacije iz detektorja. Pozorni smo še posebej na fizikalno zanimive razpadne produkte npr. mezonov B . Dogodke, ki se zapišejo in shranijo, lahko v grobem razdelimo v tri skupine. Prva skupina so za fiziko najbolj zanimivi procesi, to so razpadi hadronov s kvarkom b . Nato so trajektorije oz. sledi dveh mionov, ki izvirajo iz istega skupnega vozlišča in so koristni za ocenjevanje učinkovitosti in natančnosti celotnega procesa merjenja in rekonstrukcije. V zadnjo skupino pa spadajo razpadi mezonov D , ki se uporabljajo za kalibracijo postopka identifikacije pionov in kaonov[1].



Slika 2.9: Razdelitev trenutnega HLT2 rekonstrukcijskega programa na posamezne podprograme in njihova razvrstitev glede na računsko časovno zahtevnost. Največje računsko breme predstavlja program **TrackFit**, ki izvaja natančne rekonstrukcije trajektorij delcev[18].

Trenutna ocena za učinkovitost in hitrost HLT2 prikazuje, da postopek prilaga-

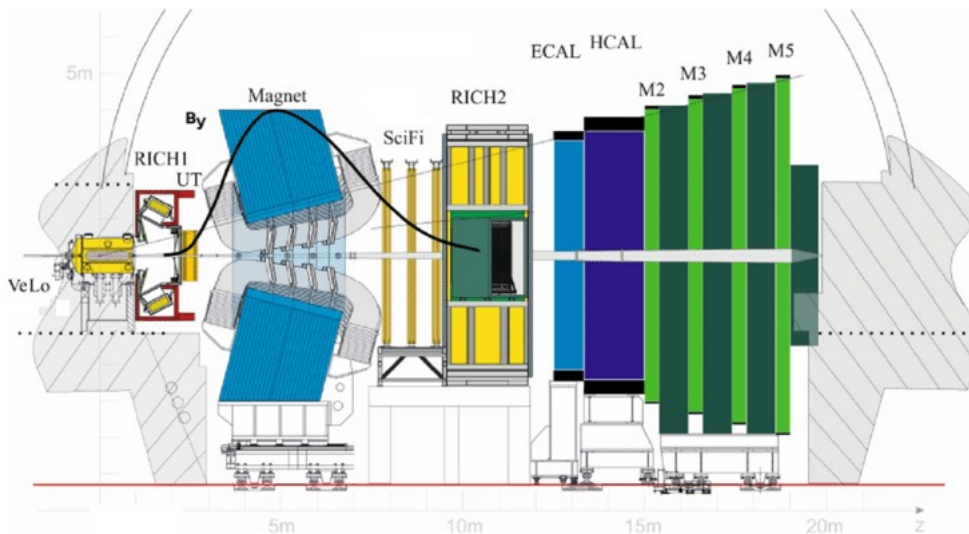


Slika 2.10: Standardni program za ekstrapolacijo delcev vzdolž njihovih trajektorij temelji na numerični integraciji Runge-Kutta algoritmov (prikazan z vijolično barvo) predstavlja 40 % bremena rekonstrukcijskega programa (ang. *TrackFit*)[19].

janja oz. določanja trajektorij zahteva 20 % celotnega računskega časa, kot lahko vidimo na sliki 2.9. Znotraj tega algoritma predstavlja 40 % tega bremena standardni program za ekstrapoliranje delcev vzdolž njihovih trajektorij, kot to prikazuje slika 2.10. To predstavlja največjo oviro pri optimizaciji rekonstrukcijske programske opreme. V naslednjem poglavju si bomo bolj natančno ogledali delovanje tega standardnega ekstrapolatorja, ki temelji na numerični integraciji tipa Runge-Kutta (RK)[19][20].

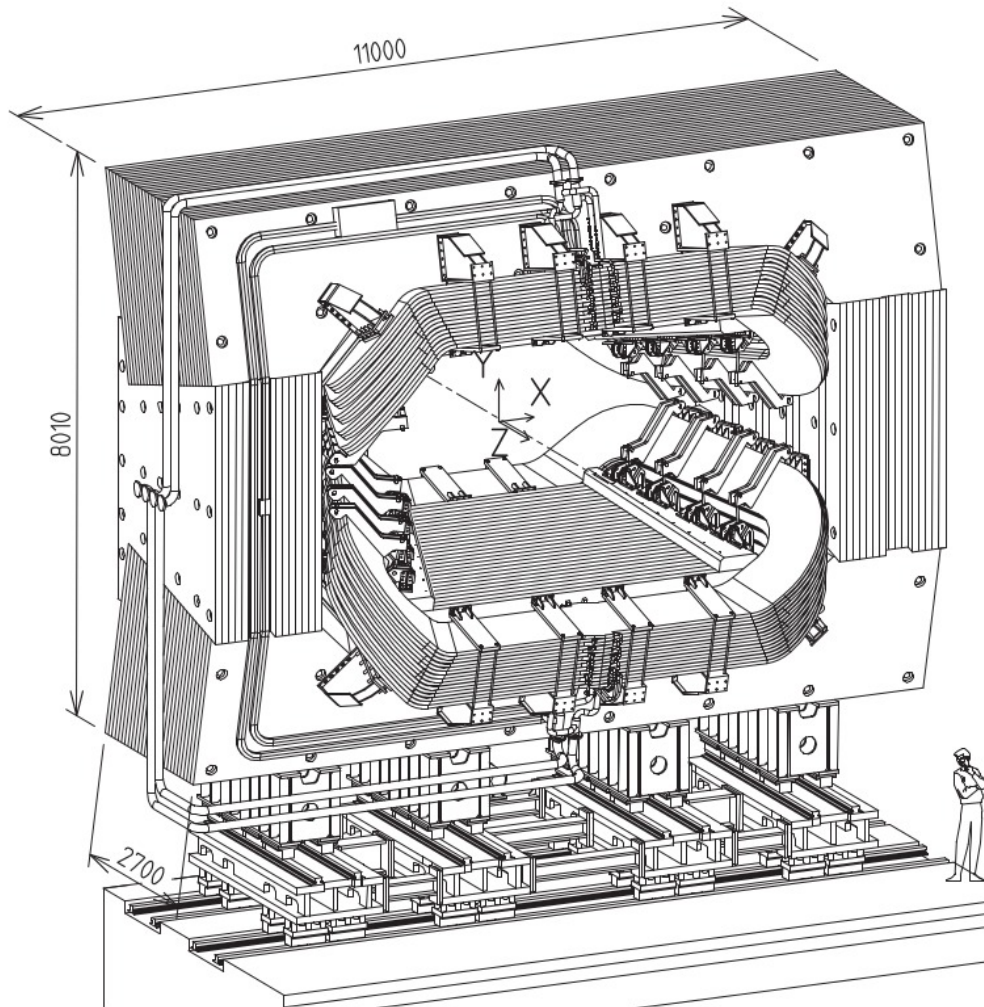
2.3 Magnetni dipol na LHCb

Detektor LHCb ima vgrajen dipolni magnet[1], ki ustvarja magnetno polje, s katerim ukrivlja trajektorije nabitih delcev in s tem, ob uporabi sledilnega sistema opisanega v poglavju 2.2.3, omogoča, da lahko izmerimo gibalno količino delcev do 200 GeV/c s približno 0.5 % natančnosti[2].



Slika 2.11: Prerez detektorja LHCb skupaj z magnetom in vsemi detektorskimi plastmi, po vrsti iz leve proti desni; VELO, RICH1, UT in *SciFi Tracker* sestavljata sledilni sistem, RICH2, kalorimetra ECAL in HCAL in na koncu mionski spektrometer. Čez magnet je narisan tudi y komponenta gostote magnetnega polja, ki doseže svoj vrh v sredini magnetu z velikostjo 1,05 T[1].

LHCb dipolni magnet je zmožen ustvariti magnetno polje z gostoto 1,05 T v smeri glavne komponentne vzdolž osi y , katere grobo krajevno odvisnost vzdolž žarkovne osi z lahko vidimo na sliki 2.11. Magnet je sestavljen iz dveh trapezoidnih tuljav znotraj pravokotnega jeklenega okvira, kot lahko vidimo na sliki 2.12. Tuljavi sta zarotirani za 45° , da sovpadeta s stožčasto obliko celotnega detektorja.

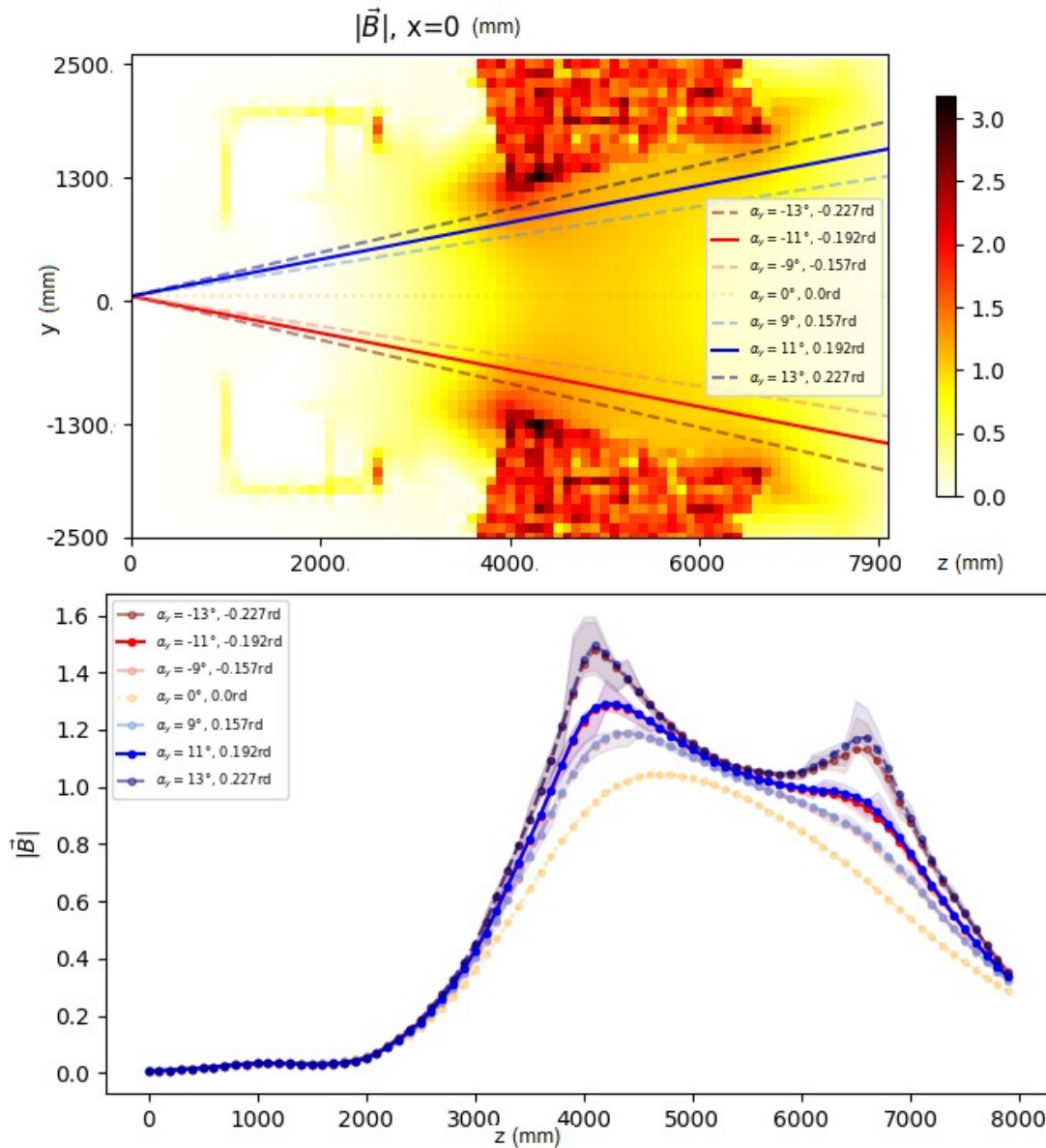


Slika 2.12: Pogled na LHCb magnet, ki prikazuje trapezoidne tuljave z jekleno sredico[1].

Rekonstrukcijski algoritmi potrebujejo za svoje delovanje informacijo o prostorski odvisnosti magnetnega polja. Ta je shranjena znotraj datotek, ki vsebujejo meritve magnetnega polja znotraj detektorja LHCb, izmerjene na preprosti tri-dimenzionalni kubični mreži. V teh datotekah je shranjenih 208.000 mrežnih točk, kar skupaj zasede okoli 30 MB računalniškega spomina⁸. Za nas je to zelo relevantno, saj kot bomo videli v naslednjem poglavju, standardni rekonstrukcijski program tipa Runge-Kutta aktivno bere vrednosti iz takih datotek. Runge-Kutta metode temeljijo na podlagi numerične integracije, ki se izkaže za časovno zamudno. Pri razvoju alternativnih metod se zato želimo izogniti numerični integraciji. Idealno bi želeli za določitev trajektorij delcev uporabiti že implementirane preproste ekstrapolatorje[21]. Te pa temeljijo ali na trivialni linearni ekstrapolaciji ali pa na prilagajanju parabolične oblike. Problem preprostih ekstrapolatorjev je v tem, da jih ne moremo uporabiti v poljubnem območju magnetnega polja. Ponavadi jih lahko uporabimo za zelo kratke ekstrapolacije (kot npr. med zaporednimi plastmi istega detektorskega sistema) v območjih kjer je magnetno polje homogeno ali pa relativno majhno[1].

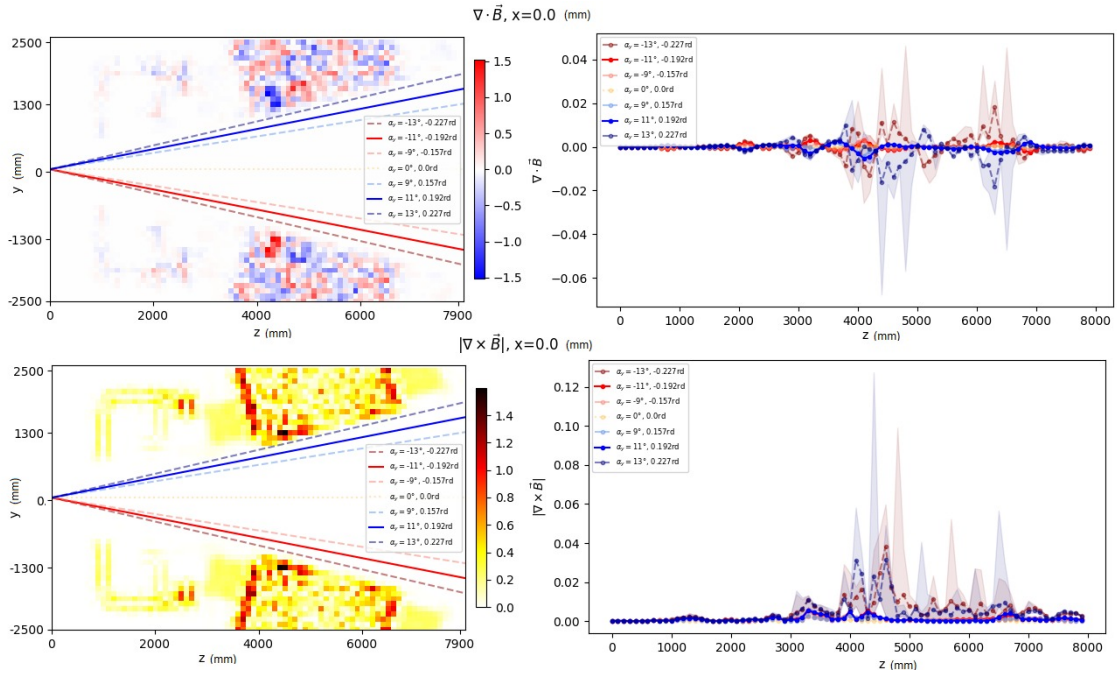
⁸Čeprav ima magnetno polje magneta na LHCb grobo obliko dipolnega polja, je prisotne veliko asimetrije, zaradi katere polja ne moremo opisati analitično.

Območje v sredini magneta zato predstavlja največji izziv. Znotraj magneta nimamo nobene detektorske plasti, zato je ekstrapolacija sledi delcev čez magnet tu najdaljša, z dolžino 5 m. Drugi problem pa je v tem, da je magnetno polje znotraj magneta zelo nehomogeno in kompleksno ter doseže zelo velike vrednosti gostote magnetnega polja. Na sliki 2.13 si lahko ogledamo prečni prerez gostote magnetnega polja znotraj najbolj kritičnega območja za ekstrapolacijo trajektorij[22].



Slika 2.13: Vizualizacija absolutne vrednosti gostote magnetnega polja znotraj magneta LHCb. Na zgornjem grafu lahko vidimo dvo-dimenzionalen prerez v xz ravnini. Očitni so artefakti, ki jih povzroča magnetna zaščita okoli RICH1 detektorja in jekleno jedro magneta. Na spodnjem grafu lahko vidimo odvisnost gostote magnetnega polja vzdolž petih črt, ki so označene na zgornji sliki. Izrazita sta dva vrhova, ki sovpadata z robovi magneta. Pet črt je bilo izbranih pri različnih prostorskih kotih za analizo najbolj optimalne akceptance, pri kateri še ne opazimo artefaktov na robu centralnega območja. Transparentni pasovi ponazarajo maksimalno napako ocenjene vrednosti magnetnega polja zaradi trilinearne interpolacije[23][24].

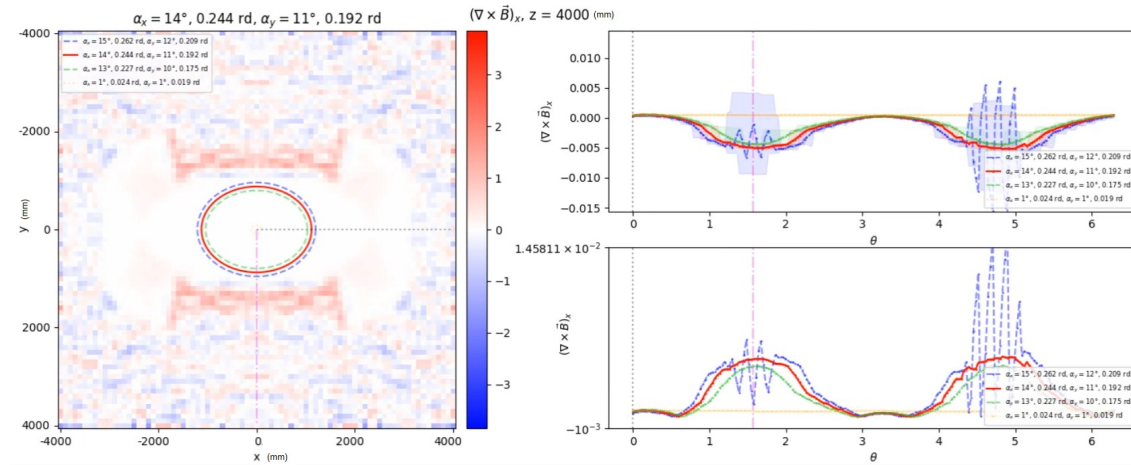
Vrednosti komponent magnetnega polja so v datotekah zapisane na podlagi diskretnih meritev. Za izračun vrednosti magnetnega polja v poljubnih točkah, ki ne ležijo točno v vozliščih na kubični mreži, se moramo zateči k interpolaciji. Kot lahko vidimo ta pristop postane problematičen za območja blizu jeklenega jedra magneta, saj se vrednosti polja spreminjajo bistveno bolj sunkovito in naletimo na problem prerodkega vzorčenja. Kljub tem efektom pa lahko vizualno prepoznamo centralno stožčasto regijo, v kateri se magnetno polje spreminja gladko. Izkaže se, da je v tem gladkem območju magnetno polje tudi *Maxwellsko kompatibilno*, torej da v tem območju izgineta divergenca in rotor gostote magnetnega polja, kot lahko tudi vidimo na sliki 2.14.



Slika 2.14: Vizualizacija divergence in absolutne vrednosti rotorja gostote magnetnega polja znotraj magneta LHCb. Na levih grafih zopet vidimo dvo-dimenzionalen prerez na xz ravnini. Tudi tukaj so očitni artefakti, ki jih povzroča magnetna zaščita okoli RICH1 detektorja in jekleno jedro magneta. Na desni strani zopet vidimo odvisnost obeh fizikalnih količin vzdolž istih petih črt, ki so označene na levih slikah. Obe količini sta zanemarljivo majhni, dokler se ne približamo robu magneta. Transparentni pasovi ponazarjajo maksimalno napako ocenjene vrednosti magnetnega polja zaradi trilinearne interpolacije[23][24].

Stožčasto obliko opišemo z dvema prostorskima kotoma. V xz ravnini definiramo kot α_x in analogno v ravnini yz kot α_y . Če izberemo prevelika kota, se preveč približamo robu magneta, zaradi česar že opazimo artefakte vzdolž roba tega centralnega območja, kot lahko vidimo na slikah 2.13, 2.14 in 2.15. Na slikah vidimo poleg vseh teh odvisnosti vzdolž roba tudi prosojne pasove, ki prikazujejo interval napake, do katere pride pri trilinearni interpolaciji magnetnega polja iz diskretne kubične mreže. Iskanje bolj preprostih ekstrapolatorskih algoritmov sledi delcev smo zaradi efektov magnetnega jedra osredotočili zgolj na to centralno stožčasto območje. Razvoj novih ekstrapolatorjev tako temelji na ideji, da želimo informacijo o vrednostih gostote magnetnega polja, ki je izvrednotena na kubični mreži, nadomestiti oz. strniti v zapis nekakšnih prostih parametrov na podlagi katerih bodo

novi ekstrapolatorji izračunali napoved trajektorije. Za optimalno izbiro prostorskih kotov, smo izbrali $\alpha_x < 0.25$ rd in $\alpha_y < 0.2$ rd⁹. K sreči to centralno območje sovпада z 96 % vseh trajektorij, do katerih tipično pride pri trkih protonov[23].



Slika 2.15: Izsek animacije odvisnosti z komponente rotorja gostote magnetne polja v xy ravnini, pri vrednosti $z = 4$ m. Na levem grafu vidimo dvo-dimenzionalen prerez na xy ravnini, pri čimer je očitno obris magnetna. Na desni strani vidimo odvisnost fizikalne količine (v linearni in logaritemski skali) vzdolž štirih elipsoid, ki so označene na levem grafu. Očitne so fluktuacije, do katerih pride pri izbiri prevelikih prostorskih kotov, ki definirata elipsoido. Prosojni pasovi ponazarjajo napako trilinearne interpolacije magnetnega polja[24][23].

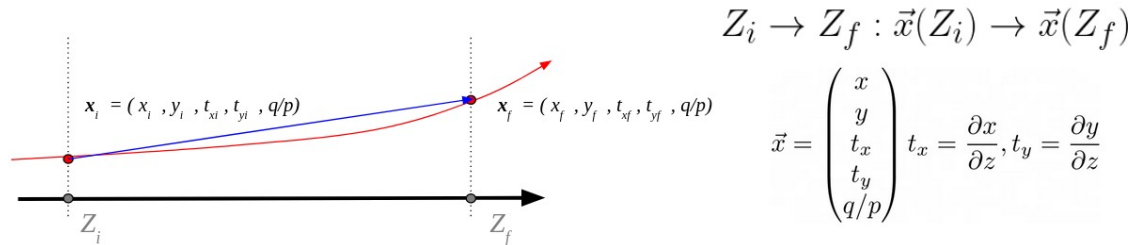
⁹To centralno območje definira sprejemno območje detektorja za merjenje trajektorij imenovano akceptanca (ang. *acceptance*).

Poglavje 3

Rekonstrukcija trajektorij nabitih delcev

Namen rekonstrukcije trajektorij, ki je zaobjeta v programski opremi pod imenom “**TrackFit**”, je izračun najbolj natančne napovedi parametrov sledi delcev skupaj z njihovimi nedoločenostmi, tipično podanih v obliki kovariančnih matrik. Te informacije se uporabijo pri meritvi gibalne količine, rekonstrukciji dogodkov in za uspešno identifikacijo delcev v kombinaciji z meritvami obročev RICH, kandidatov za mione in kalorimeterskih skupkov (ang. *cluster*). Prav tako so izračuni iz rekonstrukcije trajektorij ključni za fizikalno analizo, saj omogočajo točno določitev primarnih in sekundarnih vozlišč zanimivih fizikalnih procesov, preko česar lahko izračunamo invariantne mase delcev[2][20].

Trajektorijo opišemo s pomočjo segmentiranih daljic, ki so tangentne na sled delca. Te posamezne segmente imenujemo **vektorji stanj**. Ker vse sledi delcev lahko ločimo glede na to ali potujejo v smeri ali pa v nasprotni smeri žarkovne osi z , je naravna parametrizacija vektorjev stanj vzdolž koordinate z . Vsak vektor stanja je definiran s prostorskimi koordinatami in tangentno smerjo sledi delca pri neki specifični vrednosti z . Iz tega dobimo štiri komponente, kateri dodamo še peto vrednost, definirano kot razmerje naboja delca z njegovo gibalno količino q/p , ki je sorazmerna z ukrivljenostjo sledi delca v magnetnem polju. Obravnavamo delce z enotskim električnim nabojem $q = \pm 1$. Definicijo pet-dimenzionalnega vektorja stanja lepo povzame ilustracija na sliki 3.1.



Slika 3.1: Definicija pet-dimenzionalnega vektorja stanja skupaj z ilustracijo ekstrapolacije vzdolž žarkovne osi z .

Vektorje stanj za dano trajektorijo delca lahko definiramo ob poljubni vrednosti z . V rekonstrukcijskem procesu pa nas zanimajo vektorji stanj in njihove kovariančne matrike v različnih detektorskih plasteh, kjer sovpadajo z meritvami. Najpogostejša operacija, ki jo izvajamo pri rekonstrukciji, je ekstrapolacija meritve oz. vektorja

stanja med dvema zaporednima detektorskima ravninama, ki jo lahko opišemo z zvezo propagacije

$$\vec{x}_k = f_k(\vec{x}_{k-1}) + \vec{w}_k \quad (3.1)$$

kjer smo s k označili točko, ki sovpada z detektorsko ravnino, f je ekstrapolacijska funkcija, in \vec{w}_k pa je označena napaka modela v obliki šuma¹. Za ekstrapolacijsko funkcijo lahko izberemo vrsto različnih modelov, idealno pa želimo razviti čim bolj natančno in računsko hitro funkcijo. V primeru trivialne linearne ekstrapolacije zavzame f_k preprosto obliko

$$f_k(\vec{x}_{k-1}) = F_k \vec{x}_{k-1} \quad (3.2)$$

kjer je F_k transportna matrika definirana kot

$$F_k = \begin{pmatrix} 1 & 0 & \Delta z & 0 & 0 \\ 0 & 1 & 0 & \Delta z & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \Delta z = z_k - z_{k-1} \quad (3.3)$$

Vsaka meritev v točki k nam poda informacijo o trajektoriji. Projekcijska enačba nam opiše zvezo med meritvijo m_k in z ekstrapolacijo napovedanim vektorjem stanja kot

$$m_k = h_k(\vec{x}_k) + \epsilon_k \quad (3.4)$$

kjer je h_k merska funkcija in ϵ_k merska napaka. V primeru, da detektor v točki k izmeri zgolj koordinato x trajektorije, se merska funkcija poenostavi v

$$h_k(\vec{x}_k) = H_k \vec{x}_{k-1} \quad (3.5)$$

kjer je H_k merska matrika podana kot

$$H_k = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (3.6)$$

ta pa v splošnem zavisi od detektorja in principa pridobivanja meritve. V primeru silicijevih pikselnih detektorjev, kjer izmerimo dve komponenti vektorja stanja, bi imela merska matrika dimenzijo 2×5 . Modelski šum \vec{w}_k in merski šum ϵ_k sta privzeta kot neznani naključni motnji z ničelno pričakovano vrednostjo. Njuna nedoločnost je zaobjeta v kovariančni matriki modelskega šuma $Q_k \equiv cov(\vec{w}_k)$ in kovariančni matriki merskega šuma $V_k \equiv cov(\epsilon)$ [2].

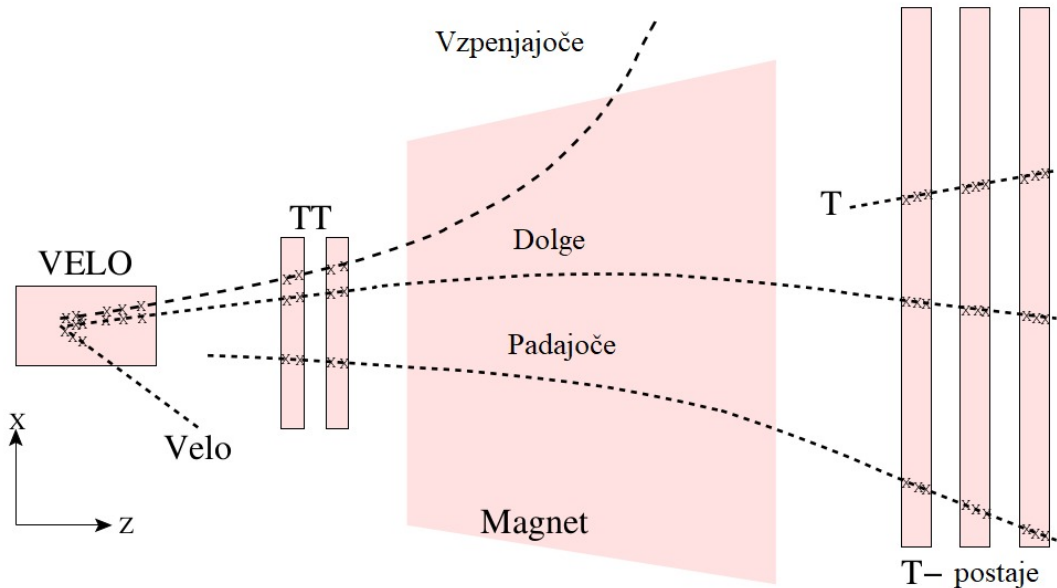
3.1 Prepoznavanje vzorcev

Rekonstrukcijski algoritmi dobijo kot vhodne podatke množico meritev oz. zadetkov iz različnih detektorskih postaj. V prvem delu rekonstrukcije moramo te zadetke modelirati v skupine in ugotoviti, kateri zadetki oz. meritve iz različnih detektorskih postaj spadajo k sledi istega delca. Pri tem si pomagamo s prepoznavanjem tipičnih karakterističnih oblik trajektorij, ki jih lahko razdelimo v naslednje tipe:

¹Ekstrapolacijske funkcije se uporabljajo tako za propagacijo delcev v smeri žarkovne osi z kot tudi za propagacijo v obratni smeri.

- **Trajektorije VELO** (ang. *VELO tracks*) vsebujejo zadetke zgolj iz detektorja VELO, tipično gre za trajektorije, ki se širijo ali v vzvratno smer glede na žarkovno os ali pa močno pravokotno glede na žarkovno os.
- **Vzpenjajoče trajektorije** (ang. *Upstream tracks*) vsebujejo zadetke zgolj pred magnetom, saj jih zaradi magnetnega polja odnese izven meritvenega območja še preden dosežejo sledilni sistem na drugi strani magnetnega polja.
- **Dolge trajektorije** (ang. *Long tracks*) imajo zadetke na vseh detektorskih postajah, zaradi česar imajo tudi najbolj določeno gibalno količino in so primarni vir podatkov za fizikalne analize.
- **Padajoče trajektorije** (ang. *Downstream tracks*) nimajo zadetkov pri detektorju VELO in so tipično posledica dovolj dolgoživih nestabilnih delcev, ki razpadejo potem, ko odletijo izven detektorja VELO.
- **T trajektorije** (ang. *T tracks*) imajo zadetke zgolj v sledilnem sistemu za magnetom in se uporabljajo za identifikacijo pionov in kaonov s pomočjo detektorja RICH2.

Vizualizacijo različnih karakterističnih tipov lahko vidimo na sliki 3.2.



Slika 3.2: Tipične karakteristične oblike trajektorij, ki jih kategoriziramo s prepoznavanjem vzorcev na začetku rekonstrukcije[1].

Prepoznavanje vzorcev temelji na analizi **Monte-Carlo** (MC) simuliranih trajektorij natančneje opisanih v poglavju 4. S pomočjo simuliranih trajektorij lahko analiziramo, kakšne zadetke tipično pustijo delci na različnih detektorskih postajah. Prepoznavanje vzorcev se začne z meritvami iz detektorja VELO, kjer vse trajektorije lahko dobro opišemo z ravno črto, in sicer zaradi majhnega magnetnega polja v tej regiji in kratke razdalje med meritvami. Ta začetni ravni segment nato služi kot osnova za ekstrapolacije vektorjev stanj v smeri naprej vzdolž žarkovne osi, s čimer lahko povežemo meritve iz zaporednih detektorjev. Pomembno je omeniti, da se veliko zadetkov izloči na podlagi različnih kriterijev, kot npr. trajektorije delcev z

gibalno količino manjšo od 1 GeV/c. Prav tako se zavrnejo trajektorije z meritvami iz sledilnega sistema, pri katerih ne moremo razlikovati zadetkov večih delcev, ki zadanejo v isto detektorsko celico hkrati[2].

Za grupiranje meritev v eno trajektorijo kot, npr. pri dolgih trajektorijah, ki so najbolj pomembne za fizikalne analize, se izvaja groba ekstrapolacija med zaporednimi detektorskimi ploščami. Izvaja se ekstrapolacija meritev v smeri vzdolž žarkovne osi z začetno aproksimacijo ravne črte pri detektorju VELO in v obratni smeri med detektorskimi postajami na drugi strani magneta. Pri vsaki zaporedni detektorski plasti se pogleda histogram odstopanj razdalj vseh zadetkov od ekstrapolirane napovedi (iz prejšnjega detektorja). Izberejo se meritve, ki najbolj sovpadajo z grobo napovedjo. Za grobo ekstrapolacijo med detektorji se uporabljajo kombinacije linearnih in paraboličnih modelov ali pa polinomskih nastavkov 3. reda.

Posebno obravnavo potrebujejo grobe ekstrapolacije skozi magnet, za kar uporabimo t.i. metodo *p-kick*[2], ki temelji na ideji, da lahko vpliv magnetnega polja opišemo kot takojšna sunkovito spremembo gibalne količine, ki jo apliciramo v sredini magneta. Teoretično je sprememba gibalne količine odvisna od integrala magnetnega polja vzdolž trajektorije delca:

$$\Delta\vec{p} = q \int d\vec{l} \times \vec{B}. \quad (3.7)$$

Glavna komponenta Δp_x poda najbolj točno oceno velikosti celotne gibalne količine in jo zapišemo s pomočjo komponent vektorjev stanja kot:

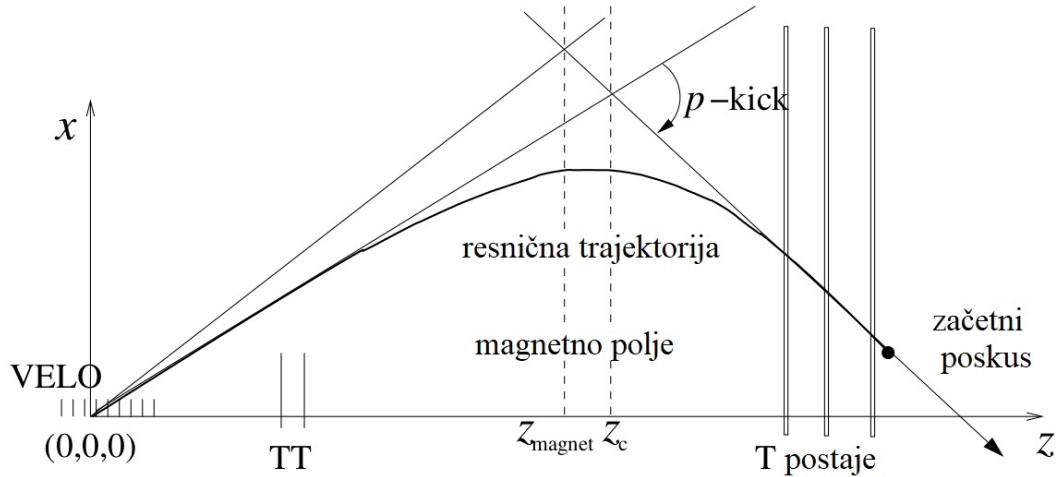
$$\Delta p_x = p_{x,f} - p_{x,i} = p \left(\frac{t_{x,f}}{\sqrt{1 + t_{x,f}^2 + t_{y,f}^2}} - \frac{t_{x,i}}{\sqrt{1 + t_{x,i}^2 + t_{y,i}^2}} \right) = q \int |d\vec{l} \times \vec{B}|_x, \quad (3.8)$$

kjer sta $t_{x,f}$ in $t_{y,f}$ naklona trajektorije v sledilnem sistemu za magnetom, ki jih dobimo na podlagi grobe parabolične interpolacije. Shematični prikaz metode *p-kick* je prikazan na sliki 3.3.

Na podlagi meritev iz sledilnega sistema za magnetom dobimo ravno črto, ki jo propagiramo nazaj do sredine magneta z_{magnet} , nato pa zavijemo proti koordinatnemu izhodišču (predpostavimo izvor točno v izhodišču). Vzdolž te začetne trajektorije iz vrednotimo integral magnetnega polja. Preko tega določimo popravljeno ravnino pri z_c , preko katere dobimo končno napoved trajektorije preko magneta. Nato spet izvedemo ekstrapolacijo v vzvratno smer skozi magnet in dobimo napoved naklona trajektorije v izhodišču $t_{x,i}$ in $t_{y,i}$ oz. pri detektorskih postajah pred magnetom². Optimalne kombinacije zaporedja meritev, ki predstavljajo posamezno trajektorijo se nato izbere na podlagi minimizacije χ^2 testa. Sledi še odstranjevanje zavrnjenih lažnih trajektorij (ang. *ghosts*) in podvojenih trajektorij, ki sovpadajo z istim delcem (ang. *clones*).

Preko začetne selekcije meritev in prepoznavanja vzorcev tipičnih trajektorij dobimo grobo rekonstruirane trajektorije, na podlagi katerih lahko učinkovito določimo in razvrstimo v skupine, kateri zadetki pripadajo prehodu istega delca. Po uspešni razvrstitvi meritev v množice trajektorij pa želimo izvesti točno rekonstrukcijo, s čimer želimo dobiti napoved trajektorije z najboljšo možno natančnostjo.

²metoda *p-kick* ima še posebno pomembno vlogo pri rekonstrukciji trajektorij, za katere nimamo meritev v sledilnem sistemu pred magnetom, torej trajektorije, ki močno zavijejo.



Slika 3.3: Shematična ilustracija metode p -kick za oceno gibalne količine trajektorije brez meritev iz detektorja VELO. Na podlagi meritev iz sledilnega sistema za magnetom se predpostavi izvor trajektorije iz izhodišča. V prvi iteraciji se predpostavi spremembo gibalne količine v sredini magneta z_{magnet} . Z integracijo magnetnega polja se napoved izboljša v ravnino z_c [2].

3.2 Kalmanov filter

Za izračun najbolj natančne rekonstrukcije trajektorij se zatečemo k uporabi **Kalmanovega filtra**[25]. Konceptualno smo problem že nastavili z množico merskih točk in z ekstrapolacijsko funkcijo, ki nam povezuje merske točke preko izračuna napovedi. S Kalmanovim filtrom iterativno korakamo vzdolž trajektorije in z vsako zaporedno meritvijo izboljšujemo oceno rekonstruirane trajektorije. Metoda temelji na minimizaciji χ^2 in je zato matematično ekvivalentna metodi najmanjših kvadratov (ang. *least-squares fit*), z razliko, da se izognemo časovno počasnemu računanju inverznih matrik³. Metoda je bila razvita okoli l. 1960 in je bila prvič uporabljena v fiziki visokih energij l. 1983 pri eksperimentu DELPHI[26]. Od takrat pa je postala standardna metoda za uporabo pri poskusih fizike visokih energij.

Kalmanov filter iterativno izvaja dva koraka: prvi korak izračuna napoved v naslednji točki, drugi korak pa na podlagi meritve v tej točki posodobi oz. popravi napovedano oceno. Kalmanov filter lahko iterira po meritvah vzdolž žarkovne osi z ali pa v obratni smeri. V detektorju LHCb se Kalmanov filter standardno izvaja v obratni smeri proti negativnim vrednostim z , s čimer dobimo najboljšo oceno trajektorije okoli izhodišča za določitev primarnih vozlišč, ki so ključnega pomena za fizikalno analizo. Pri notaciji Kalmanovega filtra bomo z $\vec{x}_{k|k-1}$ označili napoved in z $\vec{x}_{k|k} \equiv \vec{x}_k$ posodobljeno oceno. Podrobna izpeljava osnovnega Kalmanovega filtra je podana v dodatku B[2][27].

³Kalmanov filter se v resnici uporablja že pri grobi rekonstrukciji, vendar v kombinaciji s preprostejšimi ekstrapolacijskimi funkcijami, kjer se meritve dodajo k neki trajektoriji izključno na podlagi prispevka χ^2 .

3.2.1 Napoved Kalmanovega filtra

Pri izračunu napovedi izhajamo iz predhodne točke, v kateri predpostavimo, da imamo vektor stanja trajektorije skupaj z njegovo kovariančno matriko. Napoved dobimo preko propagacijske zveze:

$$\vec{x}_{k|k-1} = f_k(\vec{x}_{k-1}), \quad (3.9)$$

$$C_{k|k-1} = F_k C_{k-1} F_k^\top + Q_k. \quad (3.10)$$

Modelski šum se v obliki Q_k doda napovedi kovariančne matrike $C_{k|k-1}$. Metoda se lahko začne izvajati zgolj na podlagi neke začetne meritve oz. v praksi začetne točke iz grobe rekonstrukcije. Residual oz. odstopanje napovedi je potem podan po zvezi

$$r_{k|k-1} = m_k - h_k(\vec{x}_{k|k-1}), \quad (3.11)$$

$$R_k^{k-1} = V_k + H_k C_{k|k-1} H_k^\top. \quad (3.12)$$

Pričakovana varianca $R_{k|k-1}$ ima tako prispevek tako iz kovariančne matrike napovedi $C_{k|k-1}$ kot prispevek iz variance meritve V_k . Cilj Kalmanovega filtra je poiskati optimalno trajektorijo, ki minimizira prispevek meritve k skupni vrednosti χ^2 , ki jo izračunamo po formuli

$$(\chi_+^2)_{k|k-1} = r_{k|k-1} (R_{k|k-1})^{-1} r_{k|k-1}. \quad (3.13)$$

Podobno kot pri minimizaciji χ^2 tudi pri uporabi Kalmanovega filtra privzamemo, da sta ekstrapolacijska funkcija f_k in merska funkcija h_k linearni. Medtem ko v primeru ekstrapolacijske funkcije 3.2 in merske funkcije 3.5, to tudi res drži, operiramo v praksi z veliko bolj kompleksnimi in s tem nelinearnimi funkcijami. Uporaba kompleksnejših nelinearnih funkcij pa je pridržana za točno rekonstrukcijo. V primeru nelinearnih procesov se zatečemo k uporabi **razširjenega Kalmanovega filtra**, kjer razvijemo nelinearne funkcije v prvem redu Taylorjevega razvoja. Linearizirana ekstrapolacijska funkcija je sedaj

$$\vec{x}_{k|k-1} = f_k(\vec{x}_{k-1}^{ref}) + F_k(\vec{x}_{k-1} - \vec{x}_{k-1}^{ref}) \quad (3.14)$$

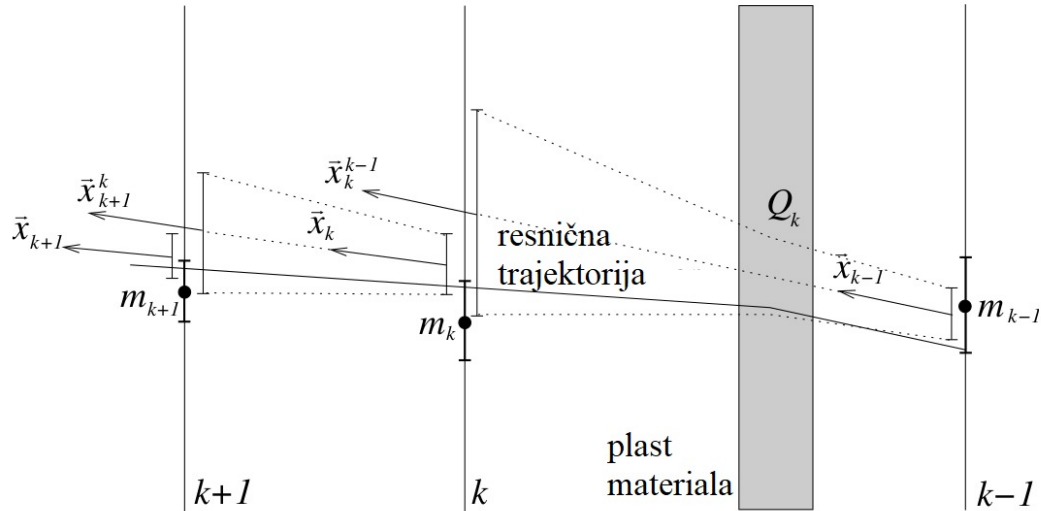
$$F_k = \frac{\partial f_k(\vec{x}_{k-1}^{ref})}{\partial \vec{x}_{k-1}^{ref}}. \quad (3.15)$$

kjer je \vec{x}_{k-1}^{ref} vektor stanja meritve iz prejšnje točke, v katero smo ekstrapolirali in filtrirali v prejšnji iteraciji. Podobno se linearizira tudi merska funkcija kot

$$h_k(\vec{x}_{k|k-1}) = h_k(\vec{x}_{k-1}^{ref}) + H_k(\vec{x}_{k-1} - \vec{x}_{k-1}^{ref}) \quad (3.16)$$

$$H_k = \frac{\partial h_k(\vec{x}_{k-1}^{ref})}{\partial \vec{x}_{k-1}^{ref}}. \quad (3.17)$$

V praksi so filtrirane posodobljene ocene vektorjev stanj že v osnovi zelo blizu pravi trajektoriji, s čimer učinkovito operiramo v linearnem režimu. Prav zaradi tega je zgolj ena iteracija dovolj za uspešno prilagajanje napovedi. V primeru nelinearnih ekstrapolacijskih in merskih funkcij moramo shemo linearizirati tudi zato, da lahko dobimo matrični izraz za transportno in mersko matriko, ki pa v prvem redu razvoja zavzameta obliko **Jacobijeve matrike**.



Slika 3.4: Shematična ilustracija delovanja Kalmanovega filtra, ki prikazuje iterativno prilagajanje meritev k optimalni oceni trajektorije. Območja gostejšega materiala povečajo napovedano napako zaradi možnega sipanja delca v snovi. Interval napake ocene je označen s črtkano črto[2].

3.2.2 Filtriranje

V drugem koraku Kalmanovega filtra posodobimo napoved vektorja stanj v novi točki na podlagi meritve iz te točke. Inovacija oz. posodobljena ocena je podana kot

$$\vec{x}_k = \vec{x}_{k|k-1} + K_k r_{k|k-1} \quad (3.18)$$

$$C_k = (\mathbb{I} - K_k H_k) C_{k|k-1} \quad (3.19)$$

kjer je *Kalmanova matrika ojačenja* podana kot

$$K_k = C_{k|k-1} H_k^\top (V_k + H_k C_{k|k-1} H_k^\top)^{-1} = C_{k|k-1} H_k^\top (R_{k|k-1})^{-1}. \quad (3.20)$$

Posodobljene vrednosti za residual in njegovo kovariančno matriko se glasijo

$$r_k = m_k - h_k(\vec{x}_{k|k-1}) = (1 - H_k K_k) r_{k|k-1}, \quad (3.21)$$

$$R_k = (1 - H_k K_k) V_k = V_k - H_k C_k H_k^\top. \quad (3.22)$$

Analogno je sedaj prispevek meritve k skupni vrednosti χ^2 :

$$(\chi_+^2)_k = r_k R_k^{-1} r_k. \quad (3.23)$$

Na sliki 3.4 lahko vidimo ilustracijo delovanja Kalmanovega filtra. Slika prikaže tudi povečanje napovedane napake modela zaradi dodatnih efektov, do katerih lahko pride zaradi sipanja delca v snovi. Kratek opis interakcije delca s snovjo je v dodatku C. Napovedane napake se izostrujejo z dodajanjem meritev v naslednjih točkah. Napoved in posodobitev ocene se iterativno izvajata dokler niso dodane vse meritve vzdolž trajektorije. Zadnja merska točka dobi s tem najbolj izostreno oceno[2].

Ker dobimo najboljšo napoved za zadnjo točko trajektorije v iteraciji Kalmanovega filtra, želimo v zadnjem delu še pogladiti napovedi v prejšnjih točkah. V

praksi se zato uporablja rekurzivna Rauch-Tung-Striebel metoda glajenja[27]. V tej shemi so poglajene ocene vektorjev stanj skupaj s svojimi kovariančnimi matrikami podane kot

$$\vec{x}_k^n = \vec{x}_k + A_k(\vec{x}_{k+1}^n - \vec{x}_{k+1|k}), \quad (3.24)$$

$$C_k^n = C_k + A_k(C_{k+1}^n - C_{k+1|k})A_k^\top, \quad (3.25)$$

kjer je 5×5 pogladitvena matrika podana kot

$$A_k = C_k F_{k+1}^\top (C_{k+1|k})^{-1}. \quad (3.26)$$

Poglajeni residual in njegova napaka sta potem

$$r_k^n = m_k - h_k(\vec{x}_k^n), \quad (3.27)$$

$$R_k^n = V_k - H_k C_k^n H_k^\top. \quad (3.28)$$

Na koncu dobimo skupno vrednost χ^2 preko vsote poglajenih prispevkov iz vseh točk:

$$(\chi_+^2)_k^n = r_k^n (R_k^n)^{-1} r_k^n. \quad (3.29)$$

3.3 Runge-Kutta ekstrapolator

Kot smo že videli v poglavju 2.3 ima eksperiment LHCb netrivialno in kompleksno magnetno polje. V regijah, kjer je magnetno polje vsaj lokalno majhno, se za ekstrapolacijo oz. napoved poti nabitih delcev zatečemo k uporabi trivialne linearne ekstrapolacijske funkcije. V območjih, kjer je magnetno polje vsaj lokalno homogeno lahko, uporabimo parabolične modele. Preprostejše ekstrapolatorje lahko uporabimo še pri ekstrapolacijah čez kratke razdalje, saj se napaka napovedi nabira postopoma z dolžino ekstrapolacije. Pri ekstrapolacijah skozi celotno območje magnetna pa ne moremo predpostaviti nobenih posplošitev o magnetnemu polju, prav tako pa so ekstrapolacije skozi magnet zelo dolge (5 m), zaradi česar se ne moremo zateči k preprostejšim in s tem hitrejšim ekstrapolatorjem.

Trenutni standardni pristop za dolge ekstrapolacije čez kompleksno magnetno polje uporablja ekstrapolatorsko funkcijo tipa Runge-Kutta (RK). Gre za iterativno metodo, ki postopoma po korakih izvaja numerično integracijo enačb gibanja za nabite delce v magnetnem polju. V odsotnosti efektov snovi, so te enačbe popolnoma definirane z *Lorentzovo silo*:

$$\vec{F} = q(\vec{E} + \vec{v} \times \vec{B}), \quad (3.30)$$

kjer je q naboj delca in v njegova hitrost. Predpostavimo, da električnega polja znotraj detektorja ni. Enačbe gibanja lahko prepisemo v odvisnosti od koordinate z s pomočjo geometrijskih količin v obliko[21]

$$\frac{d^2x}{dz^2} = \frac{q}{p} \frac{ds}{dz} \left[\frac{dx}{dz} \frac{dy}{dz} B_x - \left(1 + \left(\frac{dx}{dz} \right)^2 \right) B_y + \frac{dy}{dz} B_z \right], \quad (3.31)$$

$$\frac{d^2y}{dz^2} = \frac{q}{p} \frac{ds}{dz} \left[-\frac{dx}{dz} \frac{dy}{dz} B_y + \left(1 + \left(\frac{dy}{dz} \right)^2 \right) B_x - \frac{dx}{dz} B_z \right]. \quad (3.32)$$

kjer je q naboj delca, p njegova gibalna količina, s dolžina poti in \vec{B} lokalni vektor gostote magnetnega polja. Faktor q/p se skupno uporablja za opis ukrivljenosti trajektorije delca. Enačbe gibanja nato prepisemo v sistem navadnih diferencialnih enačb s pomočjo LHCb formalizma vektorjev stanja z modelskimi parametri $(x, y, t_x, t_y, q/p)$ v končno obliko:

$$\frac{dx}{dz} = t_x \quad (3.33)$$

$$\frac{dy}{dz} = t_y \quad (3.34)$$

$$\frac{dt_x}{dz} = \frac{q}{p} \sqrt{1 + t_x^2 + t_y^2} \left[t_y (t_x B_x + B_z) - (1 + t_x^2) B_y \right] \quad (3.35)$$

$$\frac{dt_y}{dz} = \frac{q}{p} \sqrt{1 + t_x^2 + t_y^2} \left[-t_x (t_y B_y + B_z) + (1 + t_y^2) B_x \right] \quad (3.36)$$

Te enačbe je potrebno numerično integrirati vzdolž koordinate z , s čimer propagiramo delec, da pridobimo napoved vektorja stanja v naslednji merski točki. Podrobna izpeljava enačb gibanj je opisana v dodatku A. Opazimo lahko, da imamo zgolj štiri enačbe gibanj, medtem ko je vektor stanja pet-dimenzionalen, to pa je zato, ker predpostavimo, da se energija delca ohrani (in s tem velikost gibalne količine)⁴[1][28].

Runge-Kutta numerično integracijo lahko formalno opišemo na naslednji način. Iterativno korakamo po koordinati z in pri tem propagiramo vektor stanja $\vec{x} = (x, y, t_x, t_y, q/p)$ kot:

$$\vec{x}_{n+1} = \vec{x}_n + \Delta z \sum_{i=1}^s b_i \vec{k}_i, \quad (3.37)$$

$$\vec{k}_i = f^* \left(z_n + c_i \Delta z, \vec{x}_n + \Delta z \sum_{j=1}^{i-1} a_{ij} \vec{k}_j \right), \quad (3.38)$$

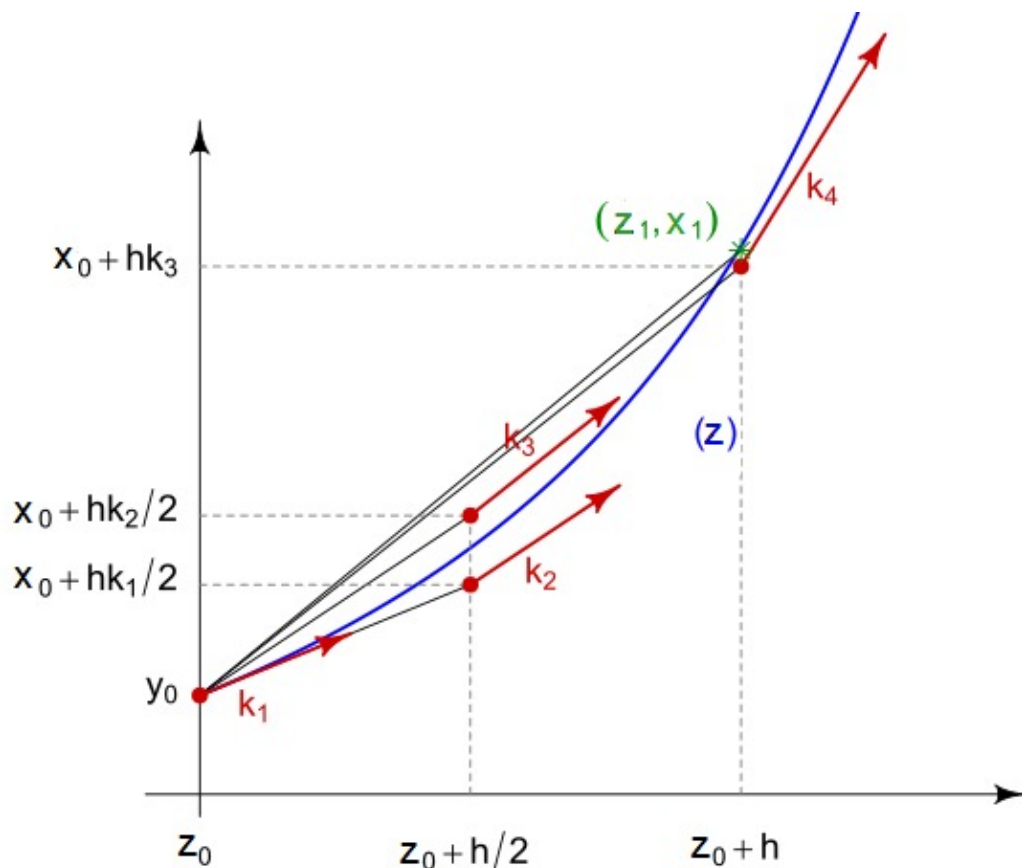
kjer je $s - 1$ red metode, oz. s velikost tabele, ki vsebuje koeficiente numerične sheme. S $f^*(z, \vec{x}) = \frac{d\vec{x}}{dz}$ smo označili funkcijo odvodov komponent vektorja stanja, ki jih imamo podane preko enačb gibanj. Metoda efektivno začne z začetno napovedjo, nato pa $s - 2$ kratno popravi oceno odvoda na danem diskretnem koraku s čimer izboljšuje napoved. Vizualizacija delovanja RK metode je ilustrirana na sliki 3.5. Standardna RK ekstrapolacija pri rekonstrukciji na LHCb temelji na numerični shemi *Cash-Karp*, ki je petega reda. Koeficiente numerične sheme prikladno zapišemo v obliki **Butcher**-jevih tabel. Za shemo *Cash-Karp* so vrednosti koeficientov zapisane v tabeli 3.1[1].

Z zadnjo vrstico koeficientov iz Butcherjevih tabel pa lahko izračunamo napoved, ki je še enega reda višja. Preko nje lahko ocenimo napako naše osnovne napovedi po formuli:

$$\vec{e}_{n+1} = \Delta z \sum_{i=1}^s (b_i - b_i^*) \vec{k}_i. \quad (3.39)$$

V primerih, ko z napovedano vrednostjo dobimo tudi ocenjeno napako, ki je večja od natančnosti, ki jo zahtevamo pri rekonstrukciji, lahko na podlagi ocene napake znova

⁴gibalna količina delca se spremeni zgolj zaradi izgube energije pri prehodu skozi snov, kot opisano v dodatku C



Slika 3.5: Ilustrativna vizualizacija delovanja klasične RK metode četrtega reda. Metoda začne z osnovno napovedjo, nato večkratno izračuna odvod funkcije vzdolž intervala korakanja, s čimer na koncu izboljša osnovno napoved[29].

Tabela 3.1: *Butcher*-jeva tabela koeficientov za numerično shemo *Cash-Karp*, ki je petega reda[1][21].

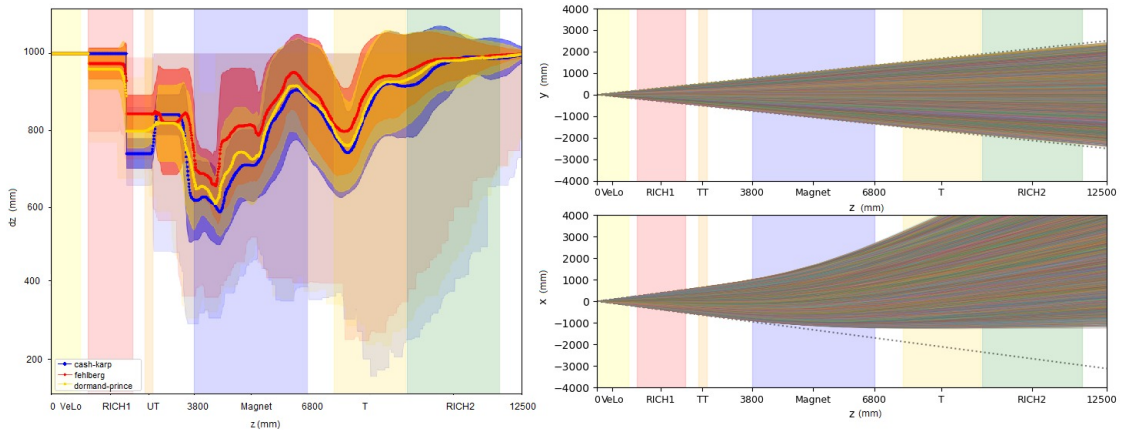
c_i	a_{ij}					
0						
$\frac{1}{5}$	$\frac{1}{5}$					
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$				
$\frac{3}{5}$	$\frac{3}{10}$	$-\frac{9}{10}$	$\frac{6}{5}$			
1	$-\frac{11}{54}$	$\frac{5}{2}$	$-\frac{70}{27}$	$\frac{35}{27}$		
$\frac{7}{8}$	$\frac{1631}{55296}$	$\frac{175}{512}$	$\frac{575}{13824}$	$\frac{44275}{110592}$	$\frac{253}{4096}$	
b_i	$\frac{37}{378}$	0	$\frac{250}{621}$	$\frac{125}{594}$	0	$\frac{512}{1771}$
b_i^*	$\frac{2825}{27648}$	0	$\frac{18575}{48384}$	$\frac{13525}{55296}$	$\frac{277}{14336}$	$\frac{1}{4}$

poskusimo zadnjo iteracijo, tokrat z manjšim korakom kot je Δz . Na podoben način lahko povečamo korak, če je ocenjena napaka zelo majhna. Na tak način sestavimo *adaptivno* RK metodo, kjer aktivno spreminjamo velikost koraka iteracije.

Implementacija adaptivne RK metode na LHCb vsebuje specifične kriterije, na

podlagi katerih prilagaja velikost iterativnega koraka, da zagotovi želeno natančnost. Za podrobnejšo analizo tega obnašanja smo implementirali LHCb standardni ekstrapolator v Python-skem peskovniku, izvorna koda je dostopna na [30]. Pri analizi smo se osredotočili na trajektorije nabitih delcev, ki izhajajo iz izhodišča in imajo gibalno količino v režimu $p \in (2, 5, 100)$ GeV/c. Standardni ekstrapolator koraka z maksimalno velikostjo koraka 1 m oz. 1000 mm. V primeru delcev z veliko gibalno količino pride zgolj do minimalne prilagoditve velikosti koraka, saj so te trajektorije trivialne po svoji obliki (blizu ravne črte).

Do izbire najmanjšega koraka pride pri delcih z majhnimi gibalnimi količinami, saj so te trajektorije tudi najbolj ukrivljene. Do največje spremembe v velikosti koraka pride v območju največjega magnetnega polja. Ker numerična integracija poteka vzdolž koordinate z in ne vzdolž neke časovne osi, je potrebno pazljivo ravnanje pri trajektorijah, ki lahko zavijejo pravokotno na smer integracije. V tem primeru je potrebno zmanjševanje koraka, ali pa v skrajnem primeru celo zamenjava smeri integracije, vendar za večino trajektorij do tega ne pride. Na sliki 3.6 lahko na levem grafu vidimo porazdelitev velikosti iterativnega koraka numerične integracije. Prikazane so tri porazdelitve za tri različne numerične sheme različnih redov natančnosti. Z bravno črto je prikaza povprečna vrednost koraka, z debelejšimi pasovi je prikazana standardna deviacija porazdelitve, s transparentnimi pasovi je prikazano celotno območje velikosti korakov. Graf je bil narisan na podlagi ekstrapolacije 10.000 trajektorij. Graf prikazuje zvezno odvisnost velikosti diskretnih korakov, kar smo izračunali tako, da smo za specifično vrednost koordinate z preverili vrednosti iterativnega koraka, ki je prečkal to vrednost z . Tako gledamo za vsako vrednost z pozardelitev 10.000 vrednosti velikosti adaptivnih korakov[30][1][21].



Slika 3.6: Na desnem grafu lahko vidimo dve ravninski projekciji deset tisočih trajektorij ($p \in (2, 5 \text{ GeV}/c, 5 \text{ GeV}/c)$), ki izhajajo iz izhodišča. Zgornja projekcija je v ravnini yz pri vrednosti $x = 0$, spodnja pa je projekcija na ravnino xz pri $y = 0$. Na levem grafu si lahko ogledamo porazdelitev povprečne velikosti iterativnega koraka pri numerični integraciji adaptivne RK metode (prikazano za numerične sheme treh različnih redov natančnosti). Adaptivna metoda močno zmanjša iteracijski korak v območju magneta, kjer tudi pride do največjega ukrivljanja trajektorij nabitih delcev[30].

Poglavje 4

Simuliranje eksperimenta

Za oceno uspešnosti rekonstrukcijskih algoritmov detektorja LHCb potrebujemo podrobno in zelo natančno simulacijo eksperimenta skupaj z rekonstrukcijo. Veliko parametrov, predvsem pri določanju kriterijskih scenarijev pri prepoznavi vzorcev, je bilo natančno uglasenih na podlagi različnih simulacij. Simulacije so zelo uporabne pri primerjavi s podatki iz dejanskih meritev, kjer služijo kot dober model ozadja, na podlagi katerega lahko pri fizikalni analizi sklepamo na obstoj še ne poznanih procesov nove fizike. Simulacije pa so tudi izjemno uporabno orodje pri ocenjevanju oz. testiranju potencialnih izboljšav v programski opremi rekonstrukcijskega procesa. Simulirane podatke bomo tako namreč rekonstruirali tako s standardnimi rekonstrukcijskimi metodami kot tudi z novo implementiranimi metodami, s čimer bomo dobili dobro objektivno oceno o uspešnosti novih metod.

Simulacija trkov visokoenergijskih delcev je sestavljena iz dveh delov: generiranja dogodkov in simuliranja odziva detektorjev. V prvem delu simuliramo iskane fizikalne procese ob trku protonov. Generiramo dogodke, pri katerih nastanejo novi delci. Če so novo nastali delci nestabilni, simuliramo tudi njihove razpade. Generiranje dogodkov je končano, ko nam ostanejo samo še stabilni delci, to so delci, ki so dovolj dolgoživi, da ne razpadejo znotraj detektorja. Informacijo o nastalih stabilnih delcih predamo simulaciji odziva detektorja. V drugem delu simuliramo, kako se delci gibljejo skozi naš detektor. Prehod delcev simuliramo po korakih. Med prehodom lahko prihaja tudi do razpadov in sipanj v materialu detektorja. Simuliranje detekcije je zaključeno, ko se vsi delci ustavijo ali uidejo iz detektorja.

Pri simulacijah procesov, pri katerih imamo opravka z osnovnimi delci, se nahajamo v režimu kvantnih pojavov. Narava, kot jo trenutno poznamo, je polna stohastičnih procesov. Če želimo simulirati take pojave, moramo torej biti sposobni ponarediti to naključnost. Medtem ko pravi generatorji naključnih števil ne obstajajo, bomo predpostavili, da obstaja tak, ki je za naše potrebe dovolj dober. Imamo torej generator, ki nam vrača realna števila R na intervalu med 0 in 1 z enakomerno verjetnostno porazdelitvijo. Dobljena števila so med seboj popolnoma nekorelirana. Pri poljubnih porazdelitvah si pomagamo z Monte Carlo (MC) metodami. Najboljše bi bilo seveda uporabiti *metodo inverznega vzorčenja*[31], vendar za to potrebujemo integrabilne porazdelitve. V takih primerih so bolj uporabne metode tipa *zadeni in zgreši* (ang *hit & miss*)[31] in *pomembnostno vzorčenje* (ang *importance sampling*)[32]. K numeričnim metodam pa se zatečemo tudi pri integralih, ki jih med drugim potrebujemo za izračune sipalnih presekov in razpadnih širin[31].

Pri delovanju LHC pride do nastanka ogromne množice dogodkov. Čeprav je

večina dogodkov nezanimivih za fizikalno analizo, jih moramo vključiti v simulacije za potrebe uglaševanja prožilnega sistema in drugih rekonstrukcijskih metod. Tem večinskim standardnim procesom, iz katerega nastane večina trajektorij delcev, pravimo **dogodki z minimalno pristranskostjo** (ang. *minimum bias events*). Ime se nanaša na nepristransko delovanje prožilnega sistema, če bi izbiral naključne dogodke. Za fiziko so najbolj zanimivi B razpadni kanali za raziskovanje kršitve CP simetrije, vendar je verjetnost, da bo prišlo do teh procesov relativno majhna v primerjavi z ostalimi procesi. Tako se v grobem simulacije v eksperimentu LHCb delijo na tri tipe: simulacije z nepristranskimi dogodki, dogodki ki vsebujejo $b\bar{b}$ pare in dogodke signala (kjer kot signal imenujemo specifični fizikalni proces, ki nas zanima)[2].

4.1 Pregled simulacije

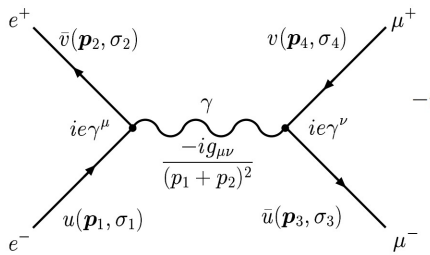
Programska oprema na LHCb temelji na objektno orientiranem programskem orodju imenovanem Gaudi[33]. Gre za programsko ogrodje napisano v programskem jeziku C++ in vključuje vse osnovne operacije od rekonstrukcije, simulacije do fizikalne analize podatkov. Procese iz programskega paketa lahko zajamemo v naslednjih grobih tipih:

- **Generiranje dogodkov.** Dogodki se začnejo generirati s simulacijo trkov dveh protonov. Simulacija pp trkov poteka s programom Pythia[31]. Na koncu tega koraka dobimo seznam izhodnih delcev, opisanih z njihovimi vektorji četverci gibalne količine. Razpadi hadronov B so posebej simulirani z aplikacijami EvtGen in Gauss[34].
- **Simulacija odziva detektorja.** V naslednjem koraku poteka simulacija poti delcev skozi detektorske plasti. Za to je zadolžen programski paket Geant4[35]. Poskrbi za interakcije delcev z materiali znotraj detektorja, za sledenje nabitih delcev v magnetnem polju in za razpad še preostalih nestabilnih delcev.
- **Digitizacija.** V tem koraku je potrebno simulirati odziv elektronike v detektorju. Tukaj se zadetek delca z detektorsko plastjo prevede oz. digitizira v električni signal. Vsaka posamezna detektorska plast ima namenjen poseben algoritem, ki pravilno simulira njen odziv. Tipično se lahko odziv detektorske plasti prevede na dva dela, v prvem simuliramo fizikalen proces generiranja signala, kot je npr. simuliranje elektronov proti žici, v drugem delu pa odziv same elektronike.
- **Prožilni sistem in rekonstrukcija.** V tej točki podatki iz simulacije izgledajo enaki izmerjenim podatkom. V tem koraku digitiziran signal pošljemo skozi dejanski prožilni sistem, preko katerega zajemamo tudi meritve iz pravega eksperimenta. Sledi poganjanje rekonstrukcijskih algoritmov, ki poteka identično kot pri obdelavi podatkov iz eksperimenta.
- **Fizikalna analiza.** Prav tako kot pri dejanskem eksperimentu predstavljajo obdelani podatki enak vir informacij za fizikalno analizo, kot pri meritvah iz eksperimenta. Po rekonstrukciji se izvede iskanje primarnih in sekundarnih vozlišč in iz tega iskanje specifičnih fizikalnih procesov, iz česar lahko na koncu

izračunamo relevantne fizikalne količine kot so invariantne mase in razpadni časi nestabilnih delcev.

Pomemben del ogrodja **Gaudi** vsebuje vse informacije o geometriji detektorjev skupaj z informacijo o materialih, postavitvah in kalibracijah posameznih detektorjev. Podatki so zajeti v obliki **XML**, s čimer jih lahko **Gaudi** preprosto in fleksibilno posreduje vsem aplikacijam, vključno paketu **Geant4**, ki simulira prehod delcev skozi detektor[35][2].

Pri obravnavanih poskusih prihaja do interakcij med osnovnimi delci. Delci se lahko sipajo med seboj ali pa razpadejo na več delcev. Verjetnost, da bo prišlo to takšnega procesa, dobimo z *matričnimi elementi* \mathcal{M} . Matrični element za dani fizikalni proces izračunamo po Feynmanovih pravilih, ki si jih lahko vizualno predstavljamo v obliki Feynmanovih diagramov.



$$-i\mathcal{M} = [\bar{u}(\mathbf{p}_3, \sigma_3)(ie\gamma^\nu)v(\mathbf{p}_4, \sigma_4)] \frac{-ig_{\mu\nu}}{(p_1 + p_2)^2} [\bar{v}(\mathbf{p}_2, \sigma_2)(ie\gamma^\mu)u(\mathbf{p}_1, \sigma_1)]$$

$$\mathcal{M} = \frac{e^2}{(p_1 + p_2)^2} [\bar{u}_3\gamma_\mu v_4][\bar{v}_2\gamma^\mu u_1] \quad u_1 \equiv u(\mathbf{p}_1, \sigma_1)$$

Slika 4.1: Primer uporabe Feynmanovih pravil[36].

MadGraph[31][32] je eden izmed najbolj uporabljenih generatorjev za avtomatični izračun matričnih elementov in se uporablja pri simulaciji poskusov na LHC. Matrični elementi \mathcal{M} nam predstavljajo opis procesa in nam dajo zvezo med vektorji četverci začetnih in končnih stanj. Iz matričnih elementov lahko nato izračunamo:

- parcialno *razpadno širino* za razpad delca $1 \rightarrow 1' + 2' + \dots + n'$

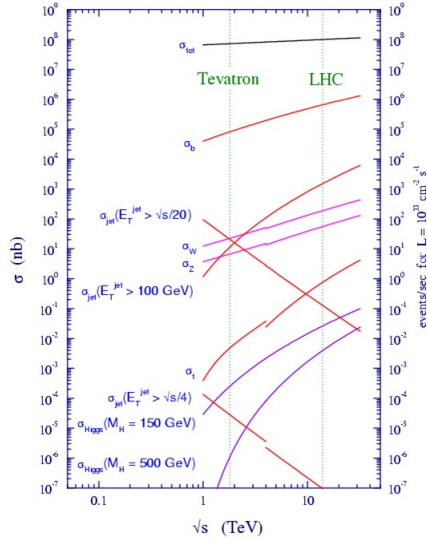
$$d\Gamma = \frac{1}{2E_1} |\mathcal{M}|^2 (2\pi)^4 \delta^4(p_1 - p'_1 - p'_2 - \dots - p'_n) \prod_{i=1}^n \frac{d^3 p'_i}{(2\pi)^3 2E'_i} \quad (4.1)$$

- ali pa diferencialni *sipalni presek* za sipanje dveh delcev $1 + 2 \rightarrow 1' + 2' + \dots + n'$

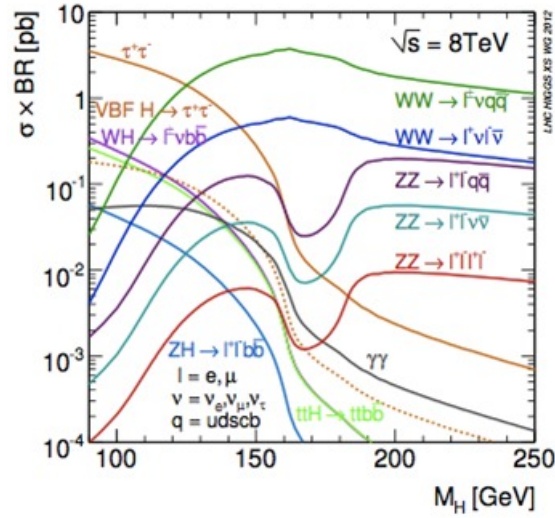
$$d\sigma = \frac{1}{4\sqrt{(p_1 \cdot p_2)^2 - m_1^2 m_2^2}} |\mathcal{M}|^2 (2\pi)^4 \delta^4(p_1 + p_2 - p'_1 - p'_2 - \dots - p'_n) \prod_{i=1}^n \frac{d^3 p'_i}{(2\pi)^3 2E'_i} \quad (4.2)$$

tukaj smo s p označeval gibalne količine posameznih delcev. Členi, ki v obeh enačbah sledijo kvadratu matričnega elementa, predstavljajo diferencial faznega prostora $d\Phi$, ki je invarianten na Lorentzovo transformacijo. Z integracijo po faznem prostoru (numerične metode) dobimo razpadno širino ali pa sipalni presek za naš proces. Če delec lahko razpade na več možnih načinov, potem je celotna razpadna širina enaka vsoti posameznih razpadnih širin. Podobno velja za sipanje delcev; celotni sipalni presek je enak vsoti posameznih sipalnih presekov. Na sliki 4.3 lahko vidimo odvisnosti različnih sipalnih presekov. Integracija po faznem prostoru pa predstavlja velik izziv za numerične metode. Z večanjem števila delcev se tudi poveča število

dimenzij faznega prostora, kar pa drastično poveča časovno zahtevnost integracije. Preko parcialnih razpadnih širin in diferencialnih sipalnih presekov dobimo tudi porazdelitve gibalnih količin novo nastalih delcev, po katerih lahko generiramo nove vektorje četverce[36][37].



Slika 4.2: Pričakovani sipalni preseki in frekvence nastajanja dogodkov nekaterih procesov SM za protonske trkalnike v odvisnosti od težiščne energije pri $L = 10^{23} \text{ cm}^{-2} \text{ s}^{-1}$ [13].



Slika 4.3: Pričakovani sipalni preseki za nekatere procese v odvisnosti od invariantne mase trka[13].

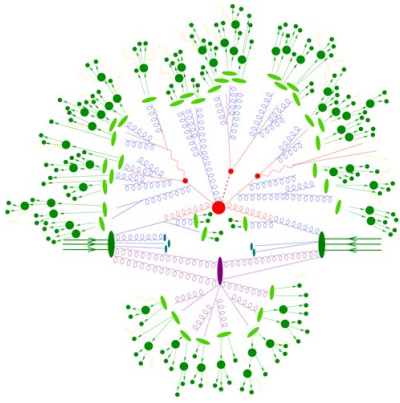
4.1.1 Generiranje dogodkov

Prej navedene enačbe za sipalni presek moramo dopolniti saj v LHC prihaja do trkov protonov, ki so sestavljeni iz kvarkov in gluonov, ki dejansko interagirajo. Pri visokoenergijskih protonih se lahko držimo *modela partonov*. Pri izračunu celotnega sipalnega preseka bomo morali upoštevati, da lahko pride do sipanja med katerem koli parom partonov. Upoštevati moramo torej sipanja med vsemi možnimi kombinacijam parov kvarkov, antikvarkov in gluonov. Sipalni presek za hadronske trke dobimo kot konvolucijo sipalnih presekov partonov in t.i. *partonske gostote*:

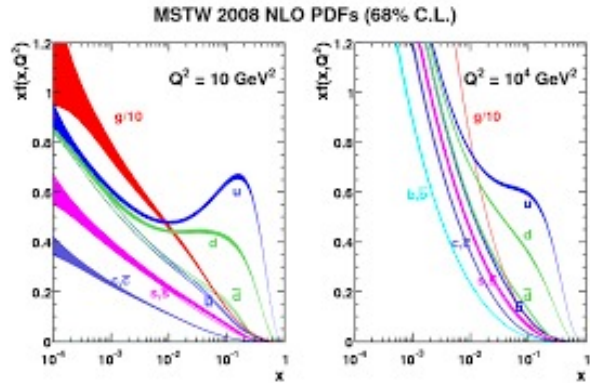
$$\sigma = \int \sum_{a,b} f_a(x_1, Q^2) f_b(x_2, Q^2) \frac{|\mathcal{M}_n|^2}{(2\pi)^{3n-4} (2x_1 x_2 s)} dx_1 dx_2 d\Phi_n \quad (4.3)$$

tukaj smo z a, b označili par partonov, s je težiščna energija trka, x_1 in x_2 sta deleža energije hadrona, ki ga nosita partona, s f pa smo označili partonski gostoti. Partonska gostota nam predstavlja popravek, ker partoni niso prosti ampak so del hadronov. Na sliki 4.2 in 4.3 vidimo odvisnost sipalnih presekov za hadronske trkalnike. V praksi so to empirično dobljene funkcije prikazane na sliki 4.5. MC generatorji običajno že vsebujejo seznam partonskih gostot.

Program za generiranje dogodkov simulira vse fizikalne procese med in po trkih protonov, do katerih pride pri naletu dveh skupkov protonov (ang. *bunch crossing*). S pomočjo programskega paketa *Pythia* simuliramo celoten proces trkov prikazan



Slika 4.4: Shema generiranja dogodka[13].



Slika 4.5: PDF: partonske gostote[13].

na sliki 4.4. Z MC metodami preko sipalnih presekov generiramo delce, ki nastanejo ob trkih hadronov. Izhajajoči kvarki in gluoni nimajo nevtralnega barvnega naboja, zaradi česar se ne morejo prosto gibati. Fenomen imenujemo zaprtje barvnega naboja¹. Precej tvorbe delcev iz kvarkov in gluonov imenujemo *hadronizacija* in je opisan fenomenološko po *Lundovem modelu strune*[31].

Na koncu je potrebno še simulirati razpade nestabilnih hadronov, do katerih je prišlo pri prej omenjenem procesu hadronizacije. Za nestabilne delce proglasimo tiste, ki razpadejo še preden bi interagirali z detektorjem. Razpade simuliramo dokler nam ne preostanejo samo še stabilni delci. Na koncu dobimo množico stabilnih delcev z njihovimi vektorji četverci. Informacije o vmesnih delcih shranimo v t.i. MC resnico, kar nam pomaga pri validaciji oz. analizi podatkov[32]. Programski paket *EvtGen* se na koncu uporablja za natančno simulacijo razpadov hadronov B , ki je ključnega pomena za LHCb[1]. Za eksperiment LHCb je ključnega pomena tvorba hadronov B . Za to morajo pri naletu skupkov protonov nastati pari $b\bar{b}$. V programu *Pythia* so štirje procesi, ki prispevajo k sipalnemu preseku tvorbe $b\bar{b}$:

1. **Tvorba para.** Pri teh procesih najnižjega reda gre za t.i. trdo sipanje. Težki okusi nastanejo pri anihilaciji kvark-antikvark parov in pri združitvi dveh gluonov, kar prispeva poglavitni delež.
2. **Vzbuditev okusa.** Virtualen kvark b iz enega protona pride na masno lupino preko sipanja s partonom iz drugega protona. Ker kvark b ni valenčen, mora obstajati \bar{b} iz predhodnega razcepa gluona.
3. **Razcep gluona.** Proces drugega najnižjega reda, pri katerem namesto trdega sipanja par $b\bar{b}$ nastane preko razcepa gluona, ki izvira iz partonskega pljuska pred ali po trdem dogodku.

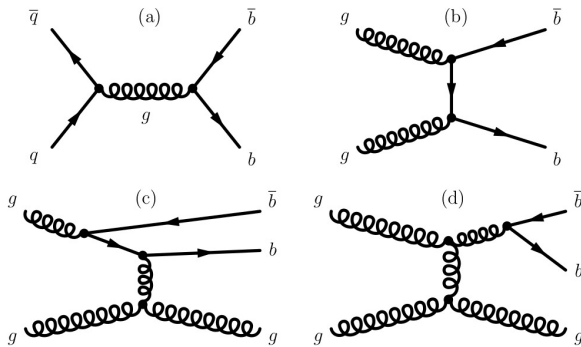
Zgoraj navedeni procesi imajo ilustrirane pridružene Feynmanove diagrame na sliki 4.6. Nastanek para $b\bar{b}$ pri procesih drugega najnižjega reda, torej vzbuditve okusa in razcepa gluona, je potrebno upoštevati kljub perturbativnemu razvoju, saj prispevata velik delež k skupnemu sipalnemu preseku. V tabeli 4.1 lahko vidimo delež, ki ga prispevajo posamezni procesi pri simulaciji s programom *Pythia*. Pri trkih

¹(ang. *color confinement*). Zaradi QCD zaprtja barvno nabiti delci namreč sevajo gluone². Močna jedrska sila hitro naraste z razdaljo, posledično se prosti kvarki vežejo v hadrone brez zunanje barvnega naboja.

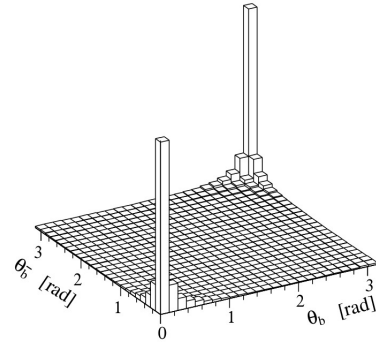
Tabela 4.1: Prispevki različnih procesov ki v simulaciji dogodkov s programom Pythia prispevajo k tvorbi hadronov B znotraj akceptance detektorja LHCb[2].

Tvorbeni proces	Prispevek (%)
(1) Tvorba para ($q\bar{q}$ anihilacija)	0.20 ± 0.04
(2) Tvorba para (združitev gluonov)	14.4 ± 0.4
(3) Vzбудitev okusa	60.3 ± 0.5
(4) Razcep gluona	25.0 ± 0.4

protonskih žarkov pričakujemo, da bo večina hadronov B , ki izvirajo iz $b\bar{b}$ parov, potovala v smeri ali obratni smeri vzdolž žarkovne osi. Pri visokih težiščnih energijah se gibalni količini protonov v trku močno razlikujeta, posledično bo nastali $b\bar{b}$ par imel močan potisk vzdolž žarkovne osi. Kotno porazdelitev hadronov B lahko vidimo na sliki 4.7[2].



Slika 4.6: Primeri Feynmanovih diagramov tipičnih za tvorbo $b\bar{b}$ parov. Procesa najnižjih redov sta (a) nastanek para preko anihilacije kvark-antikvark para in (b) združitev dveh gluonov. Procesa drugega najnižjega reda sta (c) vzбудitev okusa in (d) razcep gluona[2].

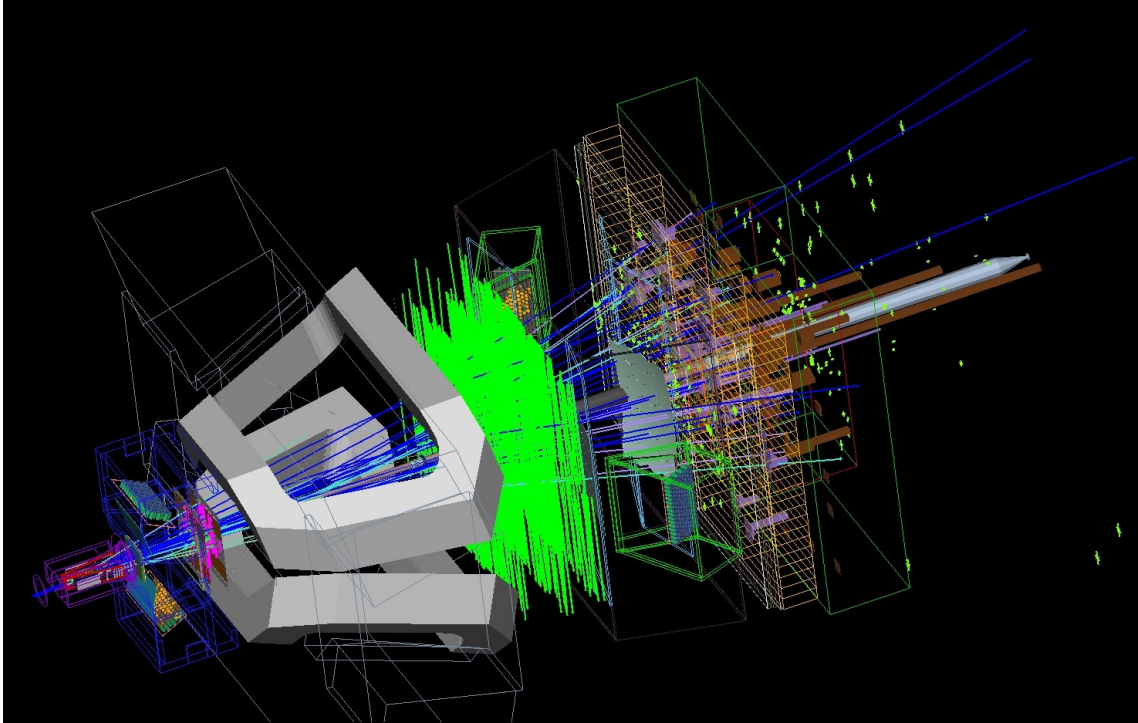


Slika 4.7: Korelacija polarnih kotov smeri trajektorije hadronov B pri težiščni energiji $\sqrt{s} = 14$ TeV [2].

4.1.2 Simuliranje odziva detektorjev

Po končani generaciji dogodkov nam preostane, da simuliramo obnašanje delcev v detektorju. Simulacijo skozi detektor lahko v grobem razdelimo na tri dele: prehod delcev skozi snov, ukrivljanje trajektorij v magnetnem polju in razpad še živečih nestabilnih delcev. Pri simulaciji tako najprej propagiramo delce skozi naš detektor. Za ta del simulacije potrebujemo natančen geometrijski opis našega detektorja. Pri tem potrebujemo natančen opis detektorskih elementov (volumnov, materialov) in še posebej natančen opis magnetnega polja znotraj našega detektorja. Simulacija prehoda delca skozi detektor poteka za vsak delec posebej. Prehod delca se simulira postopoma po korakih. Med prehodom skozi detektor lahko pride do razpada delca. Do razpadov prihaja še posebej v kalorimetrih, v katerih se zaradi teh pojavov oblikujejo prepoznavni vzorci: elektromagnetne in hadronske prhe (ang. *hadron shower*). Če delec razpade, uide iz detektorja, se absorbira oz. se ustavi in je njegova

trajektorija zaključena. Simulacija se konča ko so vse trajektorije zaključene. Med preходом delca skozi snov prihaja do različnih procesov, ki so natančno modelirani v simulacijskih paketih, kot je uporabljeni Geant[35]. Primer takšne simulacije je na sliki 4.8[12][35].



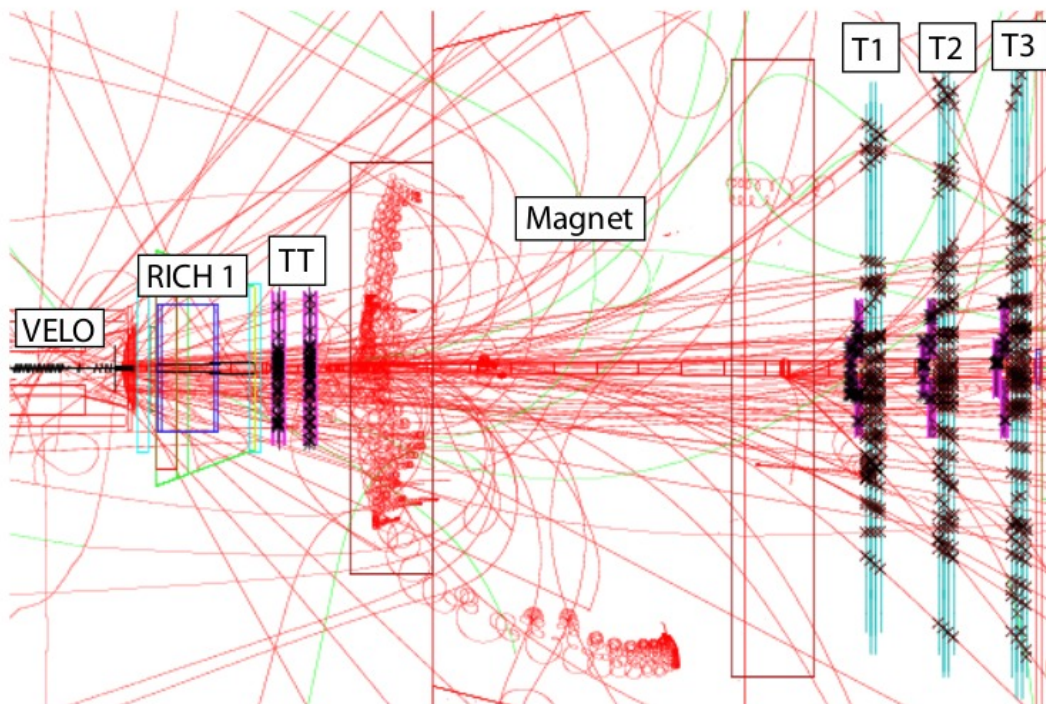
Slika 4.8: Tri-dimenzionalna vizualizacija simulacije najbolj tipičnih trajektorij pri prehodu skozi različne detektorske postaje pri eksperimentu LHCb, skupaj z ilustrirano digitizacijo signala[13].

Tipična simulacija iz programa **Geant** je prikazana na sliki 4.9. Program izvaja simulacijo delcev za vsak poddetektor posebej, kjer je potrebno upoštevati različne fizikalne procese pri npr. simulaciji Čerenkovih fotonov v RICH detektorjih ali pa pri simulaciji prh delcev v kalorimetrih[2].

Simulacija detekcije še ni končana; prehod delcev skozi detektor smo sicer že simulirali, potrebno pa je še simulirati odziv detektorja. Postopek imenujemo *digitizacija* in tu gre ponavadi za časovno zelo zahtevno simulacijo. Potrebno je upoštevati časovno odzivnost senzorjev, kar pomeni, da dobljeni signal izhaja iz večih interakcij in ne zgolj iz enega samega dogodka. Prav tako moramo upoštevati vse, kar bi prispevalo k šumu signala: stransko žarčenje curka, sevanje iz ozadja in elektronski šum na detektorjih. Iz simulacije dobimo podatke v enaki obliki, kot iz detektorja in so primerni za rekonstrukcijo ter analizo meritev. Zaradi izjemne časovne zahtevnosti polne simulacije je nesmiselno simulirati delce z nizkimi energijami, saj tipično ne dosežejo sledilnega sistema na drugi strani magnetna in so zato nezanimivi za analize fizikalnih procesov in uspešnosti rekonstrukcije dogodkov v detektorju[2].

4.2 Podatkovni seti za razvoj algoritmov

Kot že povedano, ekstrapolacije trajektorij delcev skozi magnet predstavljajo največji izziv za učinkovito in hitro rekonstrukcijo dogodkov. S simulacijami lahko



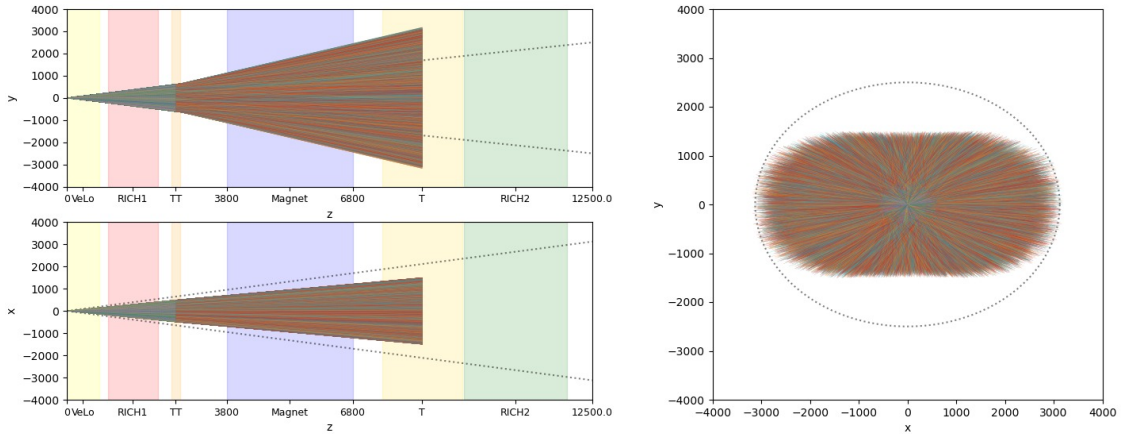
Slika 4.9: Vizualizacija simulacije odziva detektorja v programu Geant, kjer lepo vidimo ukrivljanje trajektorij nabitih delcev v magnetu detektorja LHCb in nato zadetke delcev v sledilnem sistemu na drugi strani magneta[2].

generiramo realistične podatkovne sete za uporabo učenja in prilagajanja metod, vendar pa so ti podatkovni seti lahko zasičeni s trivialnimi trajektorijami, ki jih je preprosto ekstrapolirati, ali pa po drugi strani z izjemno kompleksnimi trajektorijami, do katerih pride pri delcih z nizko gibalno količino. Pri prilagajanju modelov in oz. učenju metod strojnega učenja pa na uspešnost učenja tipično močno vpliva kvaliteta podatkovnega seta. Dober set podatkov za razvoj algoritmov za rekonstrukcijo trajektorij tako vključuje vse tipične karakteristične informacije, ki jih želimo dobro opisati, pri tem pa je ključno, da so različne karakteristike relevantno, pravilno reprezentirane v podatkovnem setu.

4.2.1 Šopkasti podatkovni set

Fazni prostor vseh karakterističnih trajektorij nabitih delcev v magnetnem polju eksperimenta LHCb je izjemno velik, v prostorskem smislu pa so le-te omejene na sorazmerno ozek segment prostorskega kota. Ta trditev se še dodatno poostri, če izločimo trajektorije delcev z nizkimi gibalnimi količinami. S temi informacijami v mislih smo zasnovali t.i. šopkasti podatkovni set (ang. *bouquet dataset*). Ime nakazuje na tipično obliko trajektorij delcev, ki jih analiziramo pri eksperimentu LHCb. Trajektorije namreč izvirajo grobo okoli izhodišča in se pri poti skozi magnet ukrivljajo pravokotno navzen od žarkovne osi, kot nekakšen šopek rož. Podatkovni set trajektorij smo generirali v parih korakih:

1. trajektorijo začnemo v izhodišču $z = 0$,
2. enakomerno smo žrebali ukrivljenost $q/p \in (1/(100 \text{ GeV}/c), 1/(2,5 \text{ GeV}/c))$,



Slika 4.10: Vizualizacija šopkastega podatkovnega seta (ang. *bouquet dataset*), ki je bil generiran z uporabo RK ekstrapolatorja s korakom 10 mm. Podatkovni set vsebuje 250.000 trajektorij[30].

3. po normalni porazdelitve smo žrebali prostorski koordinati x in y s širino $\sigma = 0.03$ mm,
4. enakomerno smo žrebali naklona $t_x \in (-0,25, 0,25)$ in $t_y \in (-0,2, 0,2)$,
5. delec ekstrapoliramo z RK ekstrapolatorjem (fiksni korak $\Delta z = 10$ mm) do $z = 2,5$ m, kjer preverimo, če zadanemo akceptanco detektorja sledilnega sistema pred magnetom (ang. *Upstream tracker*),
6. če smo zadeli detektor pred magnetom, ekstrapoliramo do $z = 7,5$ m kjer pridobimo zadnji vektor stanja, v nasprotnem primeru skočimo nazaj na točko (3).

Vizualizacijo tako pridobljenega podatkovnega seta lahko vidimo na sliki 4.10.

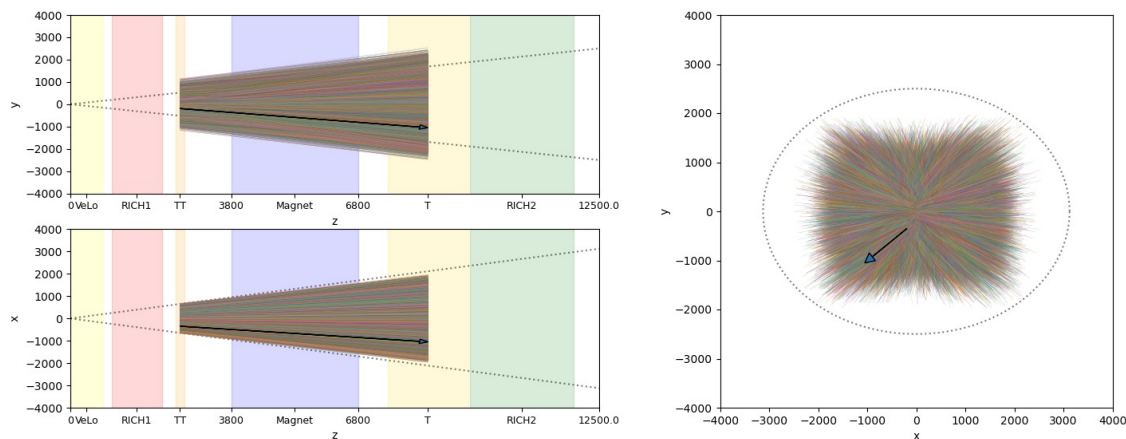
4.2.2 Podatkovni set iz simulacij

Medtem ko šopkasti podatkovni set dobro sovпада z večino realističnih trajektorij, pa se je izkazalo, da je smiselno definirati še komplementaren vzorec, analogen zajetim podatkom. Zato smo se pri prilagajanju in treniranju metod zatekli k podatkovnemu setu pridobljenemu iz simulacije procesov z minimalno pristranskostjo (ang. *minimum bias*) opisanem v poglavju 4. Gre za tipične trajektorije delcev, ki nastanejo ob trkih protonov v trkalniku. Omejimo se zgolj na dolge trajektorije, ki potujejo znotraj akceptance detektorja LHCb in zgolj za trajektorije delcev z gibalno količino večjo od 2,5 GeV/c, saj se počasnejše trajektorije tipično zavržejo, ker niso zanimive za fizikalno analizo. Vizualizacija takšnega podatkovnega seta je prikazana na sliki 4.11[2].

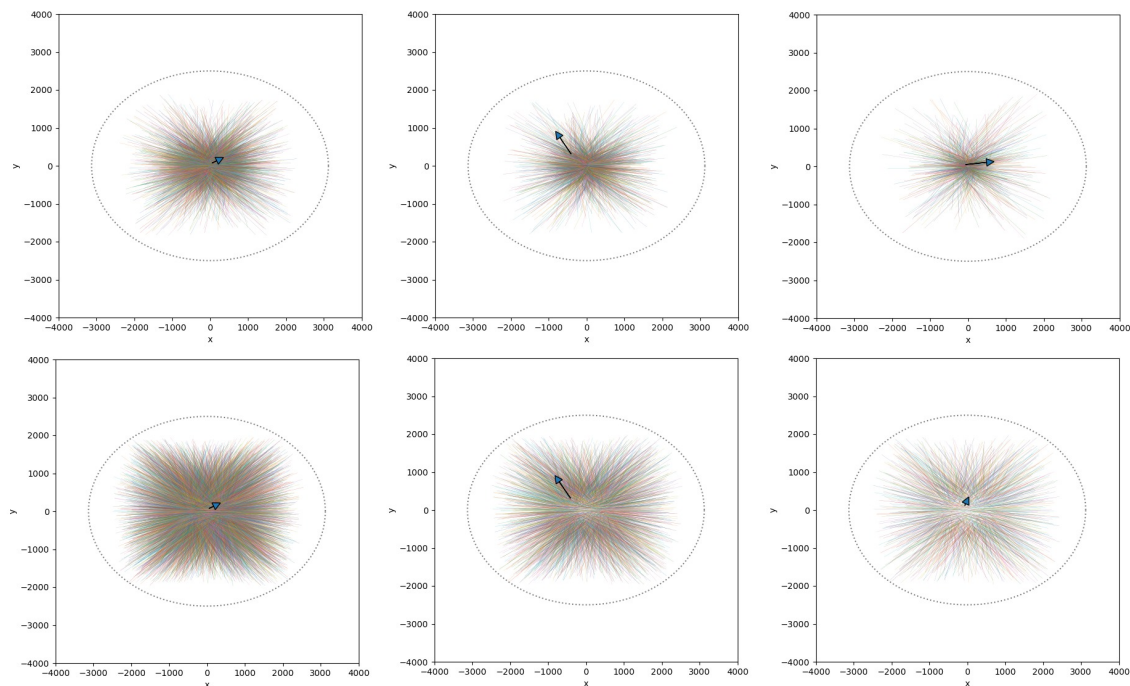
4.2.3 Kvazi-naključno vzorčenje

Pri uporabi manjše podmnožice celotnega podatkovnega seta se zopet lahko upravičeno vprašamo, ali podatkovni set enako številčno reprezentira posamezne karakteristične krivulje. Skrb je vsekakor upravičena, če samo naključno žrebamo trajektorije

Poglavje 4. Simuliranje eksperimenta



Slika 4.11: Vizualizacija podatkovnega seta pridobljenega s simulacijami z minimalno pristranskostjo (ang. *minimum bias*), ki je bil generiran z uporabo RK esktrapolatorja s korakom 10 mm. Podatkovni set vsebuje 250.000 tisoč trajektorij, od tega je izključenih 4 % trajektorij z gibalnimi količinami pod 2,5 GeV/c[2].



Slika 4.12: Vizualizacija podatkovnega seta z minimalno pristranskostjo pri različnih številih trajektorij in pri različnem načinu vzorčenja teh trajektorij. Prikazana je projekcija trajektorij na xy ravnino. Od leve proti desni podatkovni seti v obeh vrstici vsebujejo okoli 10.000, 5.000 in 2.000 trajektorij. V zgornji vrstici so bile trajektorije vzorčene enakomerno naključno, medtem ko smo v spodnji vrstici trajektorije vzorčili kvazi-naključno. Puščice označujejo smer ene naključne trajektorije iz podatkovnega seta.

iz polnega podatkovnega seta. Potencialnega problema se lahko rešimo s pomočjo kvazi-naključnega vzorčenja. Tega smo se lotili tako, da smo najprej normalizirali vektorje stanja pred in po magnetu in jih združili v en sam več-dimenzionalni vektor. Karakteristične trajektorije smo tako vzorčili, da smo naključno žrebali združen

devet-dimenzionalni vektor³ in izračunali njegovo oddaljenost do njemu najbližje že izžrebane točke kot:

$$d_{N+1} = \min_{i=0}^N (\|\vec{x}_{N+1} - \vec{x}_i\|), \quad (4.4)$$

nato smo ga sprejeli zgolj, če je bil oddaljen od njemu najbližjega že izžrebanega vektorja stanja za več kot neko vrednost r . Oddaljenost smo računali v več-dimenzionalnem Evklidijskem prostoru. Na sliki 4.12 lahko vidimo primerjavo med kvazi-naključnim in naključnim vzorčenjem podatkovnega seta z minimalno pristranskostjo. Vloga vzorčenja postane ključnega pomena pri pridobivanju reprezentativnega vzorca z majhnim številom trajektorij.

³5 komponent začetnega vektorja stanja in 4 komponente končnega vektorja stanja, skupni 9 dimenzionalni vektor vsebuje zgolj eno vrednost gibalne količine, saj smo zanemarili izgubo energije.

Poglavje 5

Nove metode za hitro ekstrapolacijo trajektorij

Ekstrapolacije trajektorij delcev so potrebne v rekonstrukcijskem procesu tako v fazi prepoznavanja vzorcev trajektorij, kot tudi pri končnem prilagajanju sledi delcev. V fazi prepoznavanja vzorcev potrebujemo natančnost, ki je reda velikosti razdalje med dvema zadetkoma delcev v detektorju. Pri končnem prilagajanju pa potrebujemo izjemno natančnost, ki mora biti nižja od vseh preostalih virov napak, ki so prisotne v napovedi. Pri končnem prilagajanju potrebujemo poleg napovedi trajektorije tudi Jacobijevo matriko vsake ekstrapolacije. Dolge trajektorije so posebej ključnega pomena za fizikalno analizo in predstavljajo veliko večino trajektorij, ki jih želimo rekonstruirati. Te prepotujejo dolgo vrzel v detektorju LHCb, kjer ne pridobimo nobene meritve o njih. To območje se nahaja točno v območju magneta, ki ustvari zelo močno in neenakomerno polje, zaradi česar nimamo preproste analitične transformacije vektorjev stanja $\vec{x} = (x, y, t_x, t_y, q/p)$ skozi območje magneta, ki meri 5 m v dolžini.

Pri ekstrapolaciji trajektorij vedno želimo poiskati najboljše ravnotežje med številom računskih operacij in količino računalniškega spomina, ki ga potrebujemo za izračun napovedi. Standardni ekstrapolator tipa Runge-Kutta aktivno bere datoteke, ki vsebujejo informacijo o magnetnem polju in zavzemajo 28,8 MB računskega spomina. Zaradi njihove adaptivne iteracijske sheme je algoritem tudi izjemno zahtevno pospešiti v smislu vektorizacije oz. paralelizacije na modernih računalniških arhitekturah. V nadaljevanju bomo predstavili tri različne pristope, s katerimi lahko relativno dobro aproksimiramo transformacijo vektorjev stanj skozi magnet in s tem hitro ter za večino trajektorij tudi zelo natančno izračunamo napoved trajektorije. Pri tem se bomo omejili na centralno območje znotraj akceptance detektorja, ki smo ga opisali v poglavju 2.3. Prav tako bomo pri prilagajanju metod izpustili trajektorije delcev z gibalno količino nižjo od 2,5 GeV/c, ker se te trajektorije tipično zavrnejo še pred končno rekonstrukcijo trajektorij[2].

5.1 Parametrizirani ekstrapolator

Medtem ko ne poznamo analitičnega zapisa za transformacijo vektorjev stanj skozi območje magneta, lahko slednjo poskusimo aproksimirati s pomočjo parametrizacije po delih. Pri začetni koordinati $z = Z_i$ si predstavljajmo dvo-dimenzionalno ploskev pravokotno na žarkovno os, po kateri razporedimo kvadratno mrežo s po-

močjo preostalih dveh koordinat x in y . Za vsako vozlišče na ploskvi želimo poiskati parametrizacijo trajektorije v odvisnosti od gibalne količine delca. Ta t.i. parametrizirani ekstrapolator je predlagal prof. Pierre Billoir[22] in je podan na podlagi naslednje formule:

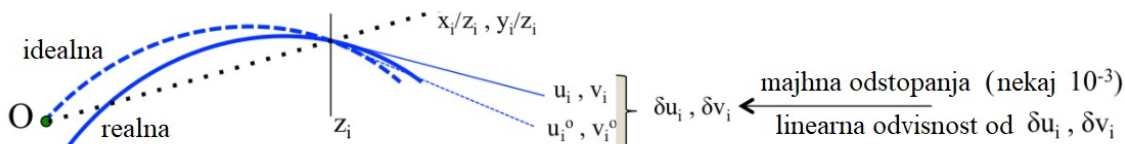
$$\vec{f}(\vec{x}_i) = \sum_{k=1}^{K_1} \vec{A}_k(x_i, y_i) \left(\frac{q}{p}\right)^k + \sum_{k=1}^{K_2} (\vec{B}_k(x_i, y_i)\delta t_x + \vec{C}_k(x_i, y_i)\delta t_y) \left(\frac{q}{p}\right)^k. \quad (5.1)$$

Zgornji izraz želimo razviti okoli ravne črte (sled delca z neskončno trajektorijo). Torej gre v bistvu za perturbativni razvoj popravkov po ukrivljenosti trajektorije. Končni izraz za posamezne trajektorije je nato za posamezne komponente vektorja stanja enak:

$$x(Z_f) = x_i + (t_x)_i(Z_f - Z_i) + \sum_{k=1}^{K_1} A_k^x(x_i, y_i) \left(\frac{q}{p}\right)^k + \sum_{k=1}^{K_2} (B_k^x(x_i, y_i)\delta t_x + C_k^x(x_i, y_i)\delta t_y) \left(\frac{q}{p}\right)^k, \quad (5.2)$$

$$t_x(Z_f) = (t_x)_i + \sum_{k=1}^{K_1} A_k^{t_x}(x_i, y_i) \left(\frac{q}{p}\right)^k + \sum_{k=1}^{K_2} (B_k^{t_x}(x_i, y_i)\delta t_x + C_k^{t_x}(x_i, y_i)\delta t_y) \left(\frac{q}{p}\right)^k. \quad (5.3)$$

Analogna izraza dobimo tudi za preostali komponenti y in t_y , medtem ko pa spremembe gibalne količine delca zaradi izgube energije ne opišemo s tem ekstrapolatorjem.



Slika 5.1: Ilustracija začetne ukrivljenosti idealnih trajektorij delcev, ki izvirajo točno iz izhodišča ($x = 0, y = 0, z = 0$). Ta ukrivljenost je do območja magneta relativno majhna in jo je potrebno upoštevati zgolj pri ekstrapolacijah z $Z_i > 2$ m. Pri tem smo uvedli krajšo notacijo preko simbolov $u = t_x$ in $v = t_y$ [38].

S tako parametrizacijo res fiksiramo prostorske koordinate in razvijemo polinomsko odvisnost od gibalne količine oz. ukrivljenosti q/p . Odvisnost od začetnega naklona trajektorije pa se skriva v členih δu in δv . Z množico prvih koeficientov \vec{A} poskušamo opisati popravek k ravni premici, s čimer skušamo modelirati t.i. *idealne trajektorije*. Za te vzamemo trajektorije delcev, ki so dosegli dano vozlišče na parametrizirani mreži in so originalno izvirali iz izhodišča ($x = 0, y = 0, z = 0$). S tem pogojem učinkovito fiksiramo tudi naklona trajektorije t_x^0 in t_y^0 , takšen naklon bi torej imele trajektorije, ki bi izvirale iz izhodišča. Večina delcev seveda ne izvira točno iz izhodišča, zato poskušamo to odstopanje modelirati v obliki razvoja okoli teh idealnih trajektorij preko koeficientov \vec{B} in \vec{C} , kjer imamo zopet polinomsko odvisnost od gibalne količine in linearno odvisnost od odstopanja naklona trajektorije

od idealnega v obliki členov $\delta t_x = t_x - t_x^0$ in $\delta t_y = t_y - t_y^0$. Začetna ukrivljenost idealnih trajektorij je tipično zelo majhna pri ekstrapolacijah bližje od $Z_i = 2$ m, čim pa se približamo magnetu, pa ju dovolj natančno opišemo z naslednjima nastavkoma:

$$t_x^0 = \xi + (G_0 + G_{xx}\xi^2 + G_{yy}\eta^2)q/p, \quad (5.4)$$

$$t_y^0 = \eta + H_{xy}\xi\eta q/p, \quad (5.5)$$

kjer smo zaradi preglednosti vpeljali spremenljivki $\xi = x_i/Z_i$ in $\eta = y_i/Z_i$. Čeprav smo parametrizirali koeficiente diskretno na dvo-dimenzionalni kvadratni mreži, po kateri so razporejeni v koordinatah $X_{n+1} = X_n + \Delta X$ in $Y_{n+1} = Y_n + \Delta Y$, smo kljub temu zapisali, kot da imajo koeficienti, npr. $\vec{A}(x_i, y_i)$, zvezno odvisnost od začetne vrednosti prostorskih koordinat. Zvezno odvisnost koeficientov \vec{A} , \vec{B} in \vec{C} aproksimiramo s pomočjo bilinearne ali bikvadratne interpolacije na kvadratni mreži, kjer za bilinearno interpolacijo vzamemo štiri najbližja vozlišča (eno kvadratno celico), pri bikvadratni interpolaciji pa šest najbližjih vozlišč. Pri uporabi bilinearne interpolacije imamo lahko težave zaradi sistematične pristranskosti (konveksnosti) interpolacije, zaradi česar se za boljšo natančnost zatečemo k bikvadratni interpolaciji. V primeru, da sta začetni vrednosti x_i, y_i najbližji vozlišču (X_n, Y_n) , definiramo šest najbližjih sosedov kot:

$$\begin{aligned} F_{00} &= (X_n, Y_n), & F_{\epsilon_x, \epsilon_y} &= F(X_n + \epsilon_x \Delta X, Y_n + \epsilon_y \Delta Y), \\ F_{+0} &= F(X_n + \Delta X, Y_n), & F_{-0} &= F(X_n - \Delta X, Y_n), \\ F_{0+} &= F(X_n, Y_n + \Delta Y), & F_{0-} &= F(X_n, Y_n - \Delta Y), \end{aligned}$$

kjer sta ϵ_x in ϵ_y predznaka spremenljivk $\zeta = (x_i - X_n)/\Delta X$ in $\psi = (y_i - Y_n)/\Delta Y$. S to notacijo lahko zapišemo interpolirano vrednost poljubnega koeficienta F kot:

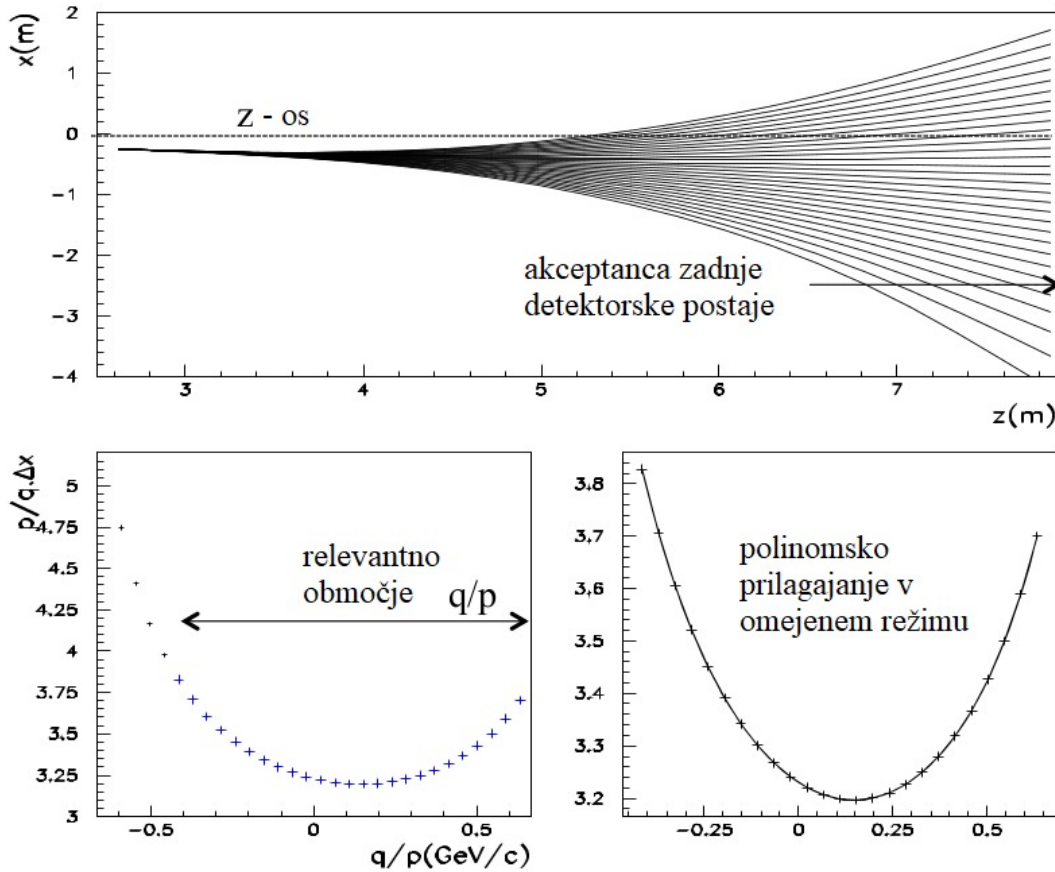
$$\begin{aligned} F(x_i, y_i) &= F_{00} + \epsilon_x \epsilon_y (F_{00} + F_{\epsilon_x, \epsilon_y} - F_{\epsilon_x 0} - F_{0 \epsilon_y}) \zeta \psi \\ &+ ((F_{+0} - F_{-0}) \zeta + (F_{+0} + F_{-0} - 2F_{00}) \zeta^2 + \\ &+ (F_{0+} - F_{0-}) \psi + (F_{0+} + F_{0-} - 2F_{00}) \psi^2) / 2. \end{aligned} \quad (5.6)$$

Metoda parametriziranega ekstrapolatorja se v splošnem lahko uporabi pri ekstrapolacijah, pri različnih izbirah detektorskih postaj lociranih vzdolž žarkovne osi ob položajih Z_i in Z_f . V naši obravnavi smo se osredotočili na najtežje ekstrapolacije skozi območje magneta med $Z_i = 2, 5$ m in $Z_f = 7, 5$ m. Kljub splošni definiciji parametrizacije moramo biti previdni glede območij ekstrapolacij, kjer dobimo dovolj dobro natančnost. Zaradi tega uporabo parametriziranega ekstrapolatorja omejimo zgolj znotraj akceptance detektorja $\alpha_x = 0, 25$ rad in $\alpha_y = 0, 2$ rad. Ravnino parametrizacije smo diskretizirali s 50×40 točkami. Polinomsko odvisnost od ukrivljenosti trajektorije smo opisali s polinomi devete stopnje, čeprav lahko za komponenti $y(Z_f)$ in $t_y(Z_f)$ uporabimo tudi nižje rede. Prav tako lahko uporabimo polinome do pete stopnje za ekstrapolacijo centralnih trajektorij, ki potujejo blizu žarkovne osi in so tipično zelo energetične z manj ukrivljenosti. Pri uglaševanju parametrizacije oz. optimiziranju prostih parametrov ocenjujemo učinkovitost metode glede na njeno natančnost. Infinitezimalno natančne napovedi ne potrebujemo, iščemo pa čim hitrejša metode, ki še vedno dovolj natančno izračunajo napoved trajektorije. Za ključno objektivno mero se sklicujemo na oceno zunanje napake (ang. *estimated external error*), ki jo opišemo z dvema členoma[22][38]:

$$\sigma_{ext}^2 = \sigma_{meas.}^2 + \sigma_{MS}^2 (|Z_f - Z_i|/p)^2, \quad (5.7)$$

Poglavje 5. Nove metode za hitro ekstrapolacijo trajektorij

kjer prvi člen opisuje prispevek merske napake, drugi pa prispevek zaradi večkratnega sipanja delcev v snovi. Slednji je seveda odvisen od snovi, hkrati pa je tudi sorazmeren s prepotovano dolžino in obratno sorazmeren z gibalno količino delca. Kot prag zadovoljive natančnosti smo vzeli desetinko te ocene.



Slika 5.2: Pri prilagajanju koeficientov parametriziranega ekstrapolatorja se omejimo zgolj na odvisnost od gibalne količine in sicer v režimu nad 2,5 GeV/c, zaradi česar odvisnosti koeficientov dobro opišemo s polinomi do devete stopnje[38].

Parametriziran ekstrapolator zaradi svoje definicije omogoča izjemno praktičen način prilagajanja več-dimenzionalne funkcije. Koeficiente \vec{A} prilagodimo zgolj na podlagi idealnih trajektorij, ki izvirajo iz izhodišča in se zato popolnoma ujemajo z oceno idealne začetne ukrivljenosti. Preostale koeficiente \vec{B} in \vec{C} pa prilagodimo tako, da vzamemo po dva para trajektorij, ki imajo rahlo pozitivno in negativno odstopanje od idealne začetne ukrivljenosti v smeri x oz. y (z razliko takih dveh parov trajektorij odpravimo odvisnost od koeficientov \vec{A}). Za vsako vozlišče (X_n, Y_n) tako izberemo pet naklonov trajektorije, da lahko nato za vsakega prilagodimo odvisnost od gibalne količine na podlagi 30 trajektorij v režimu $q/p \in (-1/(2,5 \text{ GeV}/c), 1/(2,5 \text{ GeV}/c))$, s pomočjo metode najmanjših kvadratov. Primer prilagajanja koeficientov za eno vozlišče je ilustriran na sliki 5.2. Končna parametrizacija vsebuje $4 \times 3 \times 9 \times (50 \times 40) = 216.000$ koeficientov, ki jih shranimo v datoteko velikosti 2,5 MB. Pri praktični implementaciji algoritma zaradi numeričnih napak operiramo z normaliziranimi vrednostmi komponent vektorjev stanja. Z uvedbo take parametrizacije učinkovito nadomestimo interpoliranje vrednosti vektorja gostote magnetnega

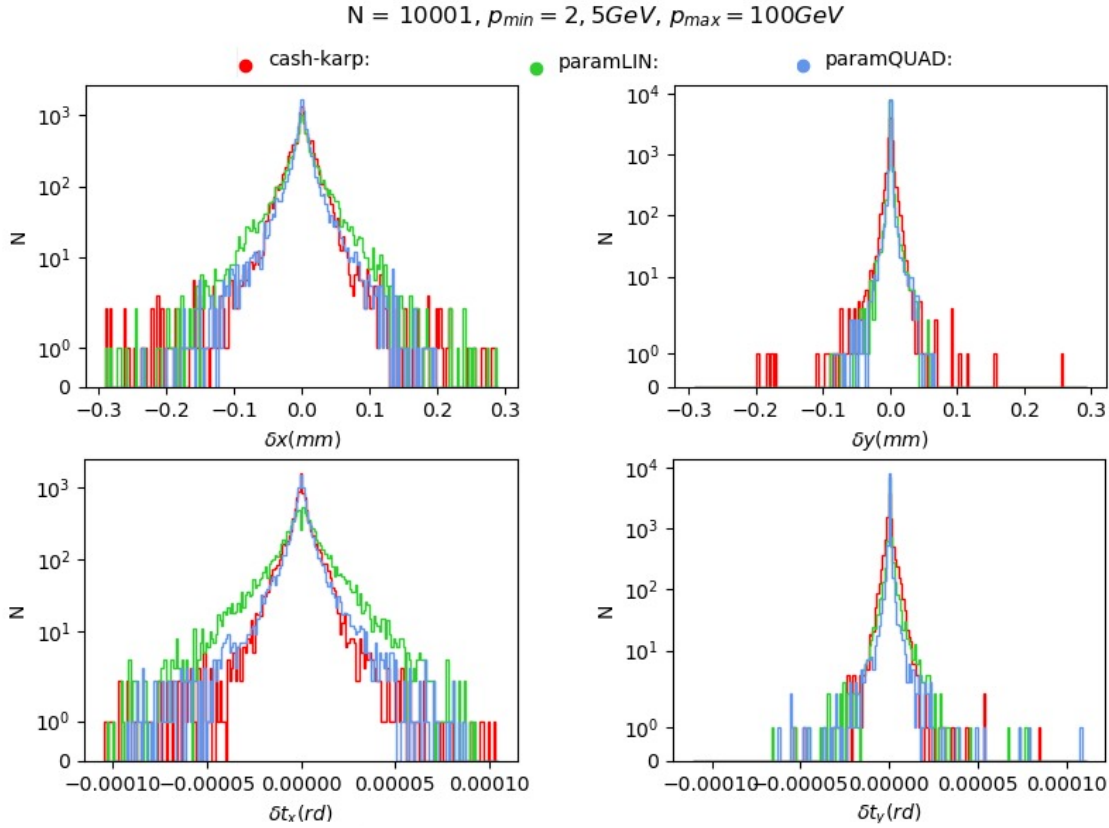
polja iz tri-dimenzionalne kubične mreže, z interpolacijo koeficientov urejenih na dvo-dimenzionalni ravnini. Tako učinkovito zmanjšamo število koeficientov za faktor tri[39].

Kot glavno orodje pri analizi metod smo uporabljali študijo, kot je prikazana na sliki 5.3. Študija je bila narejena v Python-skem peskovniku LHCb ekstrapolatorskega procesa. Ekstrapolirali smo tipične trajektorije iz podatkovnega seta z minimalno pristranskostjo skozi območje magneta v smeri naprej vzdolž žarkovne osi. Zanimajo nas predvsem odstopanja komponent končnega vektorja stanja od referenčne rešitve, kjer je slednja bila izračunana na podlagi ekstrapolatorja RK s končnim korakom 10 mm. Izgube energije nismo upoštevali. Poleg same natančnosti metode nas tipično zanima tudi njena odvisnost od gibalne količine trajektorij, saj tipično pričakujemo slabšo natančnost za netrivialne počasne trajektorije. Primer študije odvisnosti porazdelitve odstopanj od gibalne količine lahko vidimo na sliki 5.4. V splošnem si lahko privoščimo slabšo natančnost za nižje vrednosti gibalne količine, saj takrat prevladuje čedalje večji prispevek večkratnega sipanja k skupni vrednosti zunanje napake, kot lahko vidimo iz enačbe 5.7. Metodo ocenimo kot dovolj natančno, če so odstopanja od napovedi nižja od desetinke te zunanje napake podane v enačbi 5.7. Analitični izraz 5.1 prav tako preko odvajanja omogoča analitičen in zaradi tega tudi hiter izračun Jacobijeve matrike za uporabo v Kalmanovem filtru. Rezultate časovnega testiranja parametriziranega ekstrapolatorja pri ekstrapolaciji naprej skozi magnet v Python-skem peskovniku lahko vidimo v tabeli 5.1[39].

Tabela 5.1: Časovno testiranje metod za ekstrapolacije naprej skozi magnet. Rezultati so bili dobljeni na ekstrapolaciji 10.000 trajektorij iz podatkovnega seta z minimalno pristranskostjo. Časovno testiranje je bilo izvedeno v Python-skem peskovniku, ki je imitacija LHCb implementacije algoritma RK tipa Cash-Karp. Uporabljeni sta bili dve verziji parametriziranega ekstrapolatorja, prva z bilinearno interpolacijo koeficientov in druga z bikvadratno interpolacijo koeficientov. Obe verziji parametriziranega ekstrapolatorja sta vsebovali polinoma devetega reda in sta skupaj uporabljali 216.000 koeficientov parametriziranih na kvadratni mreži dimenzije 50×40 v koordinatah x_i in y_i . Časovno smo testirali zgolj čas izračuna napovedi brez izračuna Jacobijeve matrike[2][39].

Metoda	RK Cash-Karp	Par. LIN	Par. KVAD
Čas (ns)	271.793	61.225	96.670

Časovna analiza parametriziranega ekstrapolatorja naprej skozi magnet je pokazala, da metoda deluje hitreje od adaptivnega RK algoritma. Pri tem je potrebno opozoriti, da nobena izmed metod ni bila vektorizirana in smo testirali osnovno delovanje metod. Teoretična prednost parametriziranega ekstrapolatorja je v tem, da lahko, v nasprotju z adaptivno RK metodo, ekstrapolira več trajektorij hkrati na podlagi istih matematičnih operacij. Ker ima parametriziran ekstrapolator analitično formulo, je ena ekstrapolacija dovolj tudi za izračun Jacobijeve matrike[22].

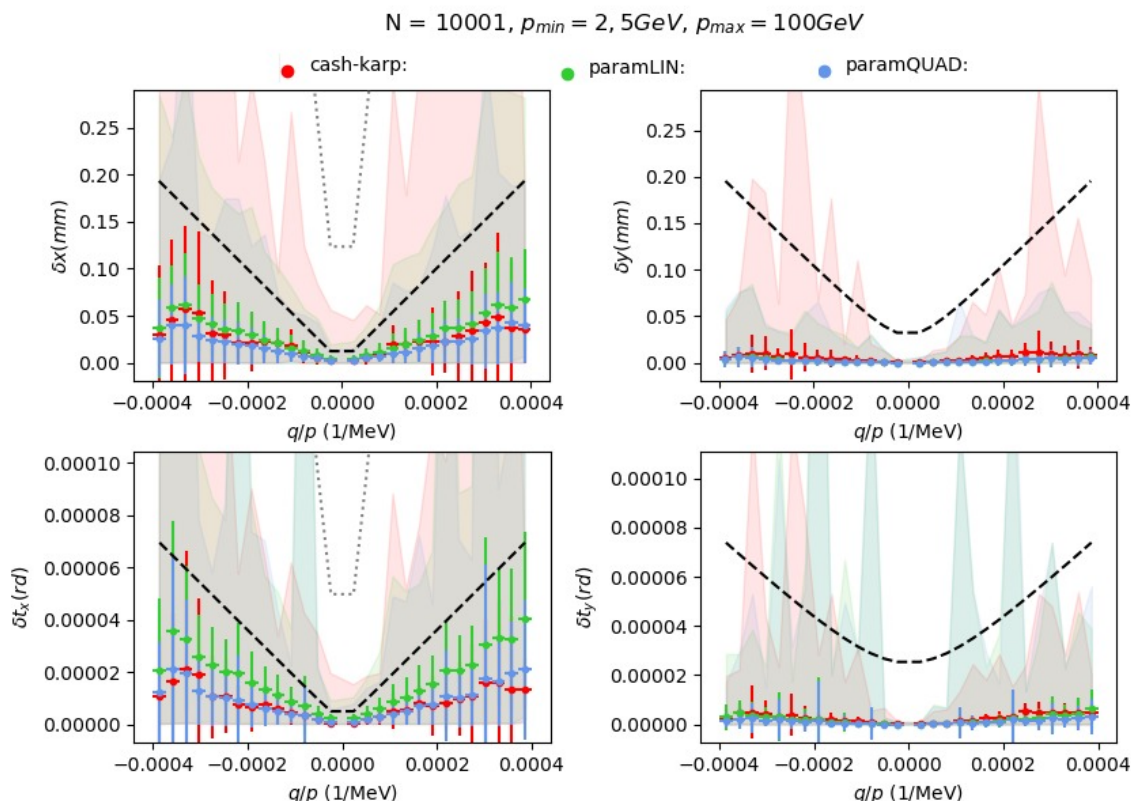


Slika 5.3: Študija natančnosti parametriziranega ekstrapolatorja (bilinearna interpolacija koeficientov = zelena, bikvadratna = modra) v primerjavi s standardnim RK ekstrapolatorjem (rdeča). Študija je bila izvedena v Python-skem peskovniku kot kopija LHCb implementacije. Uporabljen je bil podatkovni set z minimalno pristranskostjo. Prikazane so porazdelitve odstopanj napovedi komponent končnega vektorja stanja glede na referenčno rešitev izračunano z RK metodo s fiksnim korakom 10 mm[39].

5.1.1 Vzratna ekstrapolacija

Parametrizirani ekstrapolator omogoča izjemno hitro in zanesljivo ekstrapolacijo v smeri vzdolž žarkovne osi. To še zlasti drži za ekstrapolacije blizu izhodišča, ko lahko zanemarimo začetno ukrivljenost trajektorij. Ta ukrivljenost znatno naraste v območju magneta, zaradi česar izraza 5.1 ne moremo več zanesljivo uporabljati, če izvajamo ekstrapolacije iz druge strani magneta. Tipična odstopanja naklonov trajektorije so že tako velika, da nismo več v linearnem režimu. To je zlasti problematično za vzratne ekstrapolacije skozi magnet, tipično pa želimo računati ekstrapolacije vektorjev stanj ravno v obratni smeri zaradi optimalnega delovanja Kalmanovega filtra. Kljub temu se lahko zatečemo k iterativnemu pristopu tipa **Newton-Raphson** (NR). Metoda prične z neko grobo oceno končnega vektorja, tokrat pri $Z_f = 2,5$ m, in nato iz grobe napovedi normalno ekstrapolira v smeri naprej, s čimer dobimo napoved začetnega vektorja pri $Z_i = 7,5$ m, ki rahlo odstopa od dejanskega začetnega vektorja. Na podlagi tega odstopanja in s pomočjo Jacobijeve matrike iterativno popravljamo končni vektor po shemi:

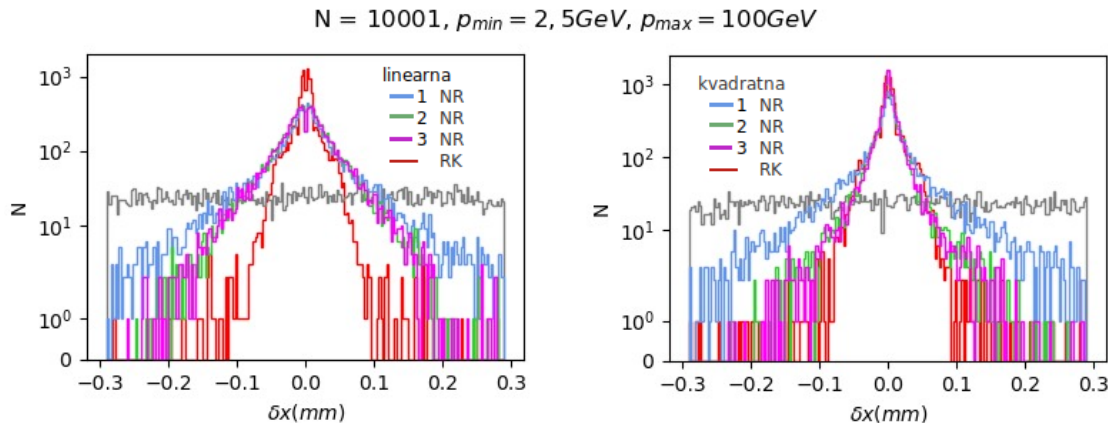
$$(\vec{x}_i)_{k+1} = (\vec{x}_i)_k + (J((\vec{x}_i)_k))^{-1}(\vec{x}_f - \vec{f}((\vec{x}_i)_k)). \quad (5.8)$$



Slika 5.4: Študija natančnosti parametriziranega ekstrapolatorja v odvisnosti od ukrivljenosti trajektorije (bilinearna interpolacija koeficientov = zelena, bikvadratna = modra) v primerjavi s standardnim RK ekstrapolatorjem (rdeča). Študija je bila izvedena v Python-skem peskovniku kot kopija LHCb implementacije. Uporabljen je bil podatkovni set z minimalno pristranskostjo. Prikazana je odvisnost širine porazdelitve odstopanj komponent napovedanega vektorja stanja od gibalne količine trajektorij. Intervali zaupanja ponazarjajo standardno deviacijo širine porazdelitve odstopanj, transparentni pasovi pa celoten interval odstopanj. S sivo krivuljo je označena ocena zunanje napake, s črno črto pa ena desetina te vrednosti[39].

Po tej shemi nadgradimo parametrizirano napoved v iterativno shemo, ki konvergira hitro, a zgolj v primeru, če že imamo tipično dobro začetno napoved končnega vektorja. V praktični situaciji pri prilagajanju končne prilagoditve trajektorij že poznamo grobo pozicijo vektorjev stanj, ki bi jih lahko uporabili kot začetne aproksimacije v Newton-Raphson shemi. Kljub temu trenutna programska arhitektura LHCb rekonstrukcijske programske opreme tega še ne podpira, zaradi česar smo se pri testiranju zatekli k Python-skemu peskovniku. Začetno grobo oceno vektorjev končnih stanj pa smo opisali na podlagi enakomerne porazdelitve okoli prave vrednosti. Rezultate vzvratnih ekstrapolacij za bilinearno in bikvadratno interpolacijo lahko vidimo na sliki 5.5.

Problem iterativnih shem je v tem, da konvergirajo pri različnem številu korakov, odvisno od začetne vrednosti. Ker pa želimo poiskati metode, ki bi se jih dalo vektorizirati oz. paralelizirati, se moramo omejiti na končno število korakov iteracije. Preko analize LHCb prožilnega sistema smo ocenili, da sta že samo dve iteraciji dovolj za zadostno konvergenco metode, čeprav moramo pri tem uporabiti bikvadratno interpolacijo koeficientov. Parametriziran ekstrapolator vsebuje nekaj kriterijev, na



Slika 5.5: Študija natančnosti parametriziranega ekstrapolatorja (bilinearna interpolacija koeficientov = levo, bikvadratna = desno) v primerjavi s standardnim RK ekstrapolatorjem (rdeča). Študija je bila izvedena v Python-skem peskovniku kot kopija LHCb implementacije. Modra je ena iteracija NR sheme, zelena dve in vijolična tri iteracije. Siva porazdelitev prikazuje odstopanje začetne grobe ocene končnega vektorja, ki smo jo replicirali na podlagi analize LHCb prožilnega sistema (ang. *Trigger*.)

podlagi katerih obupa nad specifičnimi trajektorijami in sicer v primeru, ko je trajektorija izven parametrizirane ravnine, ali pa ko naklon trajektorije preveč odstopa od naklona idealne trajektorije. Tipično smo opazili, da se to zgodi za okoli 3 % trajektorij iz podatkovnega seta z minimalno pristranskostjo, vendar pa ta odstotek naraste do 10 % pri ekstrapolacijah v vzvratni smeri skozi magnet. Časovna analiza kaže kot pričakovano, da je časovna zahtevnost parametriziranega ekstrapolatorja z NR shemo in dvema iteracijama zahtevna približno dvakrat toliko kot ena sama ekstrapolacija navadnega parametriziranega ekstrapolatorja (še majhen dodaten faktor za izračun inverza Jacobijeve matrike dimenzije 5×5) [40][39].

5.2 Prilagajanje z metodo SVD

Parametrizirani ekstrapolator deluje izjemno dobro in natančno za ekstrapolacije v smeri vzdolž žarkovne osi in je zato zlasti uporaben pri ekstrapolacijah pred magnetom. Tako ostanejo ekstrapolacije v vzvratni smeri skozi magnet proti izhodišču najbolj zahtevne. Največja pomanjkljivost parametriziranega ekstrapolatorja leži v dejstvu, da so koeficienti prilagojeni na podlagi zelo specifičnih trajektorij, ki so blizu idealnim trajektorijam iz izhodišča. Kot alternativen pristop bi si želeli zamisliti model, ki ga lahko prilagajamo na podlagi poljubnega nabora trajektorij. Zamislimo si ekstrapolator, ki za vsako komponento vektorja stanja izračuna napoved na podlagi več-dimenzionalne funkcije, v kateri pa koeficienti, ki jih želimo prilagoditi, nastopajo zgolj linearno:

$$f(Z_f) = \sum_h \sum_k \sum_l \sum_m \sum_n a_{hklmn} (x_i)^h (y_i)^k (t_x)_i^l (t_y)_i^m \left(\frac{q}{p}\right)_i^n \quad (5.9)$$

kjer smo funkcijo razvili po posameznih komponentah začetnega vektorja s polinomskim razvojem. Seveda pa potrebujemo skupno štiri take nastavke za vse kompo-

nente končnega vektorja stanja. Če imamo sedaj podatkovni set trajektorij, ki ga želimo čim bolj opisati, potem želimo minimizirati

$$\chi^2 = \sum_N (x_f - f(x_i))^2 = \|A\vec{a} - \vec{b}\|^2, \quad (5.10)$$

kjer smo z \vec{a} označili vse prilagoditvene koeficiente dolžine M , z \vec{b} pa vseh N meritev. Tako matrika A vsebuje vse produkte polinomskih členov za vsako izmed meritev in je dimenzije $N \times M$. V primeru, da rešujemo predoločen sistem ($N > M$), lahko sistem rešimo s pomočjo metode SVD¹. Dekompozicijo zapišemo kot $A = UWV^T$, kjer je U matrika z ortogonalnimi stolpci dimenzije $N \times M$, W je $M \times M$ diagonalna matrika z nenegativnimi singularnimi vrednostmi in V je ortogonalna matrika dimenzije $M \times M$. Rešitev linearnega sistema za minimizacijo χ^2 dobimo po formuli:

$$\vec{a} = \sum_{i=1}^M \left(\frac{\vec{U}_i \cdot \vec{b}}{w_i} \right) \vec{V}_i. \quad (5.11)$$

K iskanju vrednosti parametrov preko SVD smo se zatekli zaradi ogromnega števila parametrov, ki ga potrebujemo za opis funkcije petih spremenljivk. Pri tako velikem faznem prostoru se namreč ne moremo zanašati na metode numerične minimizacije, saj se tipično minimizatorji v visoko dimenzionalnih faznih prostorih hitro ujamejo v lokalne minimume. Glavna prednost SVD pristopa pa je v tem, da nam metoda vedno poda rešitev z optimalno izbiro vrednosti prostih parametrov v smislu minimizacije *najmanjših kvadratov*, vsaj glede na naš podatkovni set. Uspešnost same metode oz. interpretacija optimalne rešitve pa zato na koncu zavisi od vhodnih podatkov, torej podatkovnega seta[41][42].

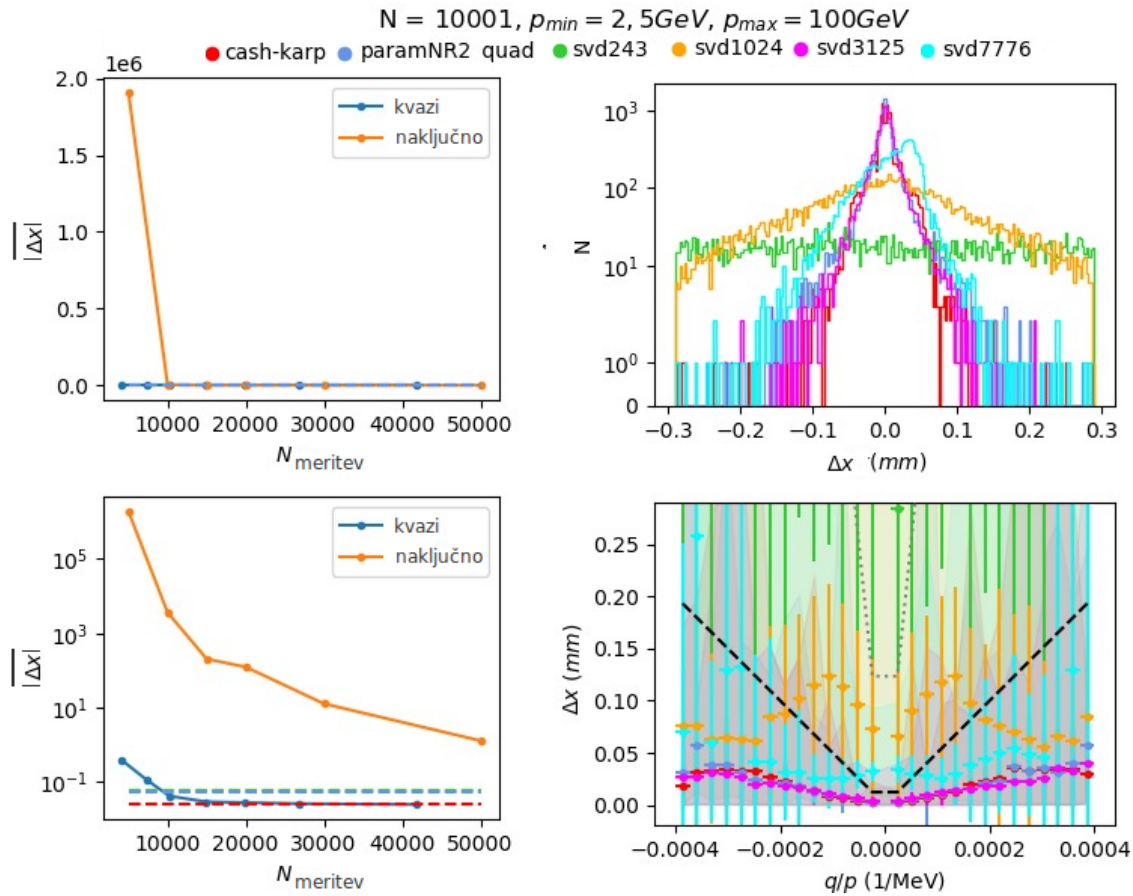
V tej točki se takoj pojavi vprašanje, koliko in katere trajektorije vključiti v nabor meritev. Za uspešno konvergenco metode smo uporabili podatkovni set z minimalno pristranskostjo, vendar pa zaradi omejitev računalniških zmogljivosti nismo uspeli vključiti celotnega podatkovnega nabora 200.000 trajektorij. Zato smo pogledali konvergenco širine porazdelitev odstopanj komponent napovedanega končnega vektorja, kjer nas je zanimalo, kako se porazdelitev ostri z dodajanjem večjega števila meritev. Kot pa se je izkazalo, je bilo še bolj ključno vprašanje, kako vzorčiti trajektorije iz podatkovnega seta. Metode učenja v podatkovnih setih tipično minimizirajo najbolj številčno populacijo trajektorij. Kot najbolj učinkovito se je izkazalo kvazi-naključno vzorčenje trajektorij, ki smo ga opisali v poglavju 4.2.3, kjer smo izbirali trajektorije, ki so si bile med seboj dovolj različne. Potek konvergence lahko vidimo na sliki 5.6.

Drugo vprašanje je bilo, koliko parametrov vključiti v prilaganje. V nasprotju s parametriziranim ekstrapolatorjem nastavek 5.9 ne temelji na neki fizikalno vodeni intuiciji, zaradi česar težko uganemo optimalen red baznih funkcij v nastavku. Zaradi tega smo pogledali konvergenco pri višanju najvišjih redov polinomskih členov v razvoju. Kot lahko vidimo na sliki 5.6, se je izkazalo, da so polinomski členi do petega reda velikosti še vedno izboljšali natančnost metode. Kljub temu ne moremo zagotovo trditi, da smo izbrali optimalno izbiro redov, saj pri prilaganju parametrov zaradi omejitev računske zmogljivosti nismo morali vključiti več kot zgolj 50.000 trajektorij (SVD matrike dimenzije $5^5 \times 50.000$). Večanje reda polinomov je na koncu začelo slabšati natančnost metode, kar smo interpretirali, kot da smo imeli

¹SVD - razcep na singularne vrednosti (ang. *singular value decomposition*.)

Poglavje 5. Nove metode za hitro ekstrapolacijo trajektorij

premajhen učni vzorec (zaradi računske omejitve velikosti matrike) in sistem 5.10 ni bil več dovolj predoločen.



Slika 5.6: Na levi sliki je prikazana odvisnost širine porazdelitve odstopanj za komponento x_f v odvisnosti od števila trajektorij uporabljenih za prilagajanje metode SVD z 2000 koeficienti in redi členov v razvoju (5, 5, 5, 4, 4) za komponente vhodnega vektorja stanja. Kvazi-naključno vzorčenje je bolj učinkovito, saj vključuje bolj reprezentativen vzorec, pri omejenem končnem številu trajektorij. Črtkane črte na levih grafih nakazujejo natančnost RK metode z rdečo in NR shemo parametriziranega ekstrapolatorja z modro. Na desni sliki lahko vidimo porazdelitve odstopanj komponente x_f pri SVD ekstrapolatorju z različnim številom prostih parametrov. Število prostih parametrov uporabljenih pri posameznem modelu je naveden na vrhu slike. Vidimo da pri omejenem številu 50.000 trajektorij iz podatkovnega seta spleča višati rede členov do petega reda, kar nam da nekje med 2000 in 3000 prostih parametrov. Nadaljno povečevanje števila parametrov na končnem podatkovnem set poslabša natančnost metode.

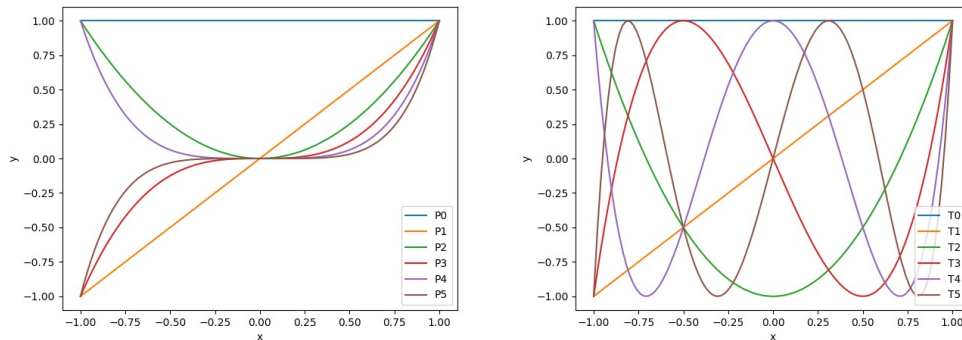
Kljub relativno dobri natančnosti modela pa se upravičeno lahko bojimo problema “pretreniranja”, torej prilagoditve parametrov na specifičen vzorec. Idealno bi želeli izbrati samo določene člene, ki so najbolj ključni za natančnost modela in bi s tem izluščili nek skriti analitičen vzorec našega problema. V praksi pa, kot tudi v našem primeru, za dinamiko našega problema nimamo analitičnega obnašanje zaradi česar t.i. luščenje odvečnih parametrov ni trivialno, ampak se izkaže kot izjemno zahteven problem. Poskusili smo namreč postopoma iterativno izločati člene s pripadajočimi parametri z visokimi variancami in oz. z visokimi relativnimi variancami

(varianca deljena z vrednostjo parametra). Izkazalo se je, da to ni bil dober pristop, saj smo izgubljali natančnost modela[41]. Pravilen pristop bi bil postopoma izločati vsak člen posebej na podlagi njegovega prispevka k χ^2 , vendar pa bi to zahtevalo ogromno število SVD dekompozicij sorazmerno z $M!$, kjer je M število parametrov pred začetkom luščenja[42]. Kljub neuspelemu luščenju pa smo namesto razvoja v normalnih polinomskih členih izbrali raje razvoj s Čebiševimi polinomi prve vrste, ki jih zapišemo po rekurzivni formuli, skupaj z njihovimi odvodi:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1} = 2xT_n(x) - T_{n-1}(x), \quad (5.12)$$

$$\frac{dT_0}{dx} = 0, \quad \frac{dT_1}{dx} = 1, \quad \frac{dT_{n+1}}{dx} = 2T_n(x) + 2x\frac{dT_n}{dx} - \frac{dT_{n-1}}{dx}, \quad (5.13)$$

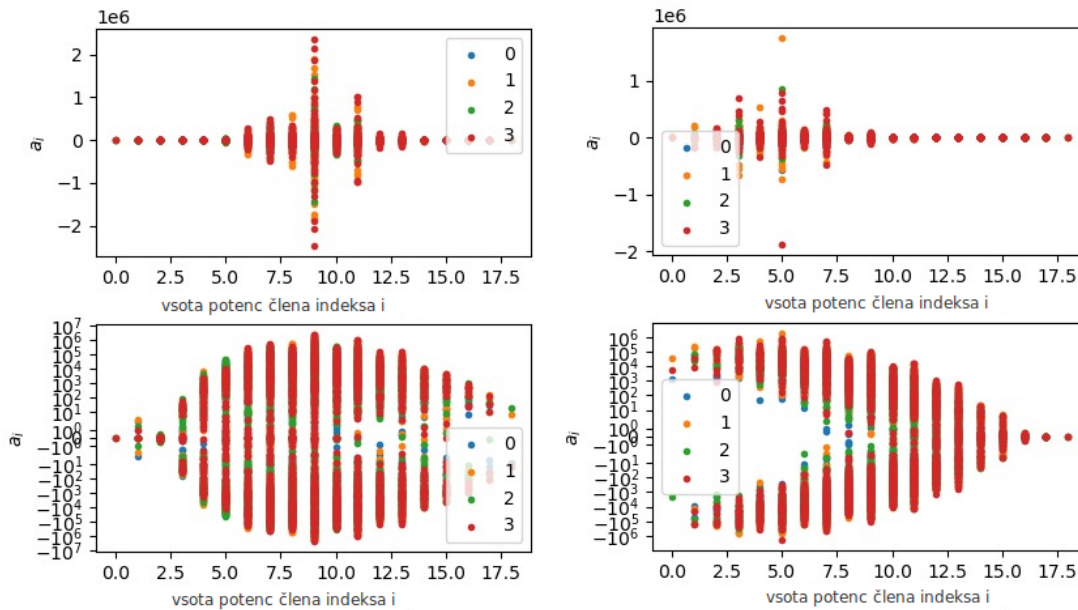
kjer potrebujemo tudi rekurzivno zvezo za odvode za izračun Jacobijeve matrike ekstrapolatorja. Čebiševi polinomi so med seboj ortogonalni in se izkažejo kot zelo praktično uporabni pri numeričnih aproksimacijah funkcij. Ilustracijo njihove oblike lahko vidimo na sliki 5.7. V našem primeru smo s pomočjo Čebiševih polinomov zagotovili, da so imeli največje vrednosti parametri nižjih redov, kar ponavadi želimo, ker želimo modelirati zglačene regularizirane funkcije. Odvisnost vrednosti parametrov od reda pripadajočega polinomskega člena lahko vidimo na sliki 5.8[42].



Slika 5.7: Na levi sliki je prikazanih prvih pet členov v navadnem polinomskem razvoju, na desni pa prvih pet členov Čebiševih polinomov prvega tipa.

Za konec je pomembno poudariti, da je v nasprotju s parametriziranim ekstrapolatorjem, pri SVD ekstrapolatorju izjemno pomembno na podlagi kakšnega podatkovnega seta prilagajamo parametre. S tem pristopom dobimo optimalno vrednost parametrov, ki minimizira χ^2 , vendar je njena vrednost odvisna od množice trajektorij. Prednost tega pristopa je v tem, da nismo omejeni na specifične trajektorije oz. specifične geometrije detektorjev, temveč lahko poiščemo optimalen model za določeno podmnožico trajektorij. V našem primeru smo se osredotočili na tipične dolge trajektorije iz protonskih trkov. Ta podatkovni set vsebuje hkrati zelo raznolike trajektorije in hkrati malo energetskih trajektorij, ki jih potrebujemo za fizikalno analizo. To neenakomerno poseljenost trajektorij smo sicer poskušali omiliti na podlagi kvazi-naključnega vzorčenja, ki se je izkazalo za zelo učinkovito. Glavno sporočilo te študije pa je, da bi lahko ta SVD pristop aplicirali tudi za drugačne tipe trajektorij in v detektorjih netrivialnih geometrij, česar ne moremo storiti s parametriziranim ekstrapolatorjem. Glede na računsko omejitev velikosti matričnega sistema 5.10, smo za optimalno število vzeli 2.000 parametrov (krat štiri za vsako komponento izhodnega vektorja), pri čemer so bili členi v razvoju 5.9 do petega reda,

kar je očitno tudi iz slike 5.6. Opravili smo tudi časovno testiranje metode, ki je navedeno na koncu naslednjega poglavja v tabeli 5.2.

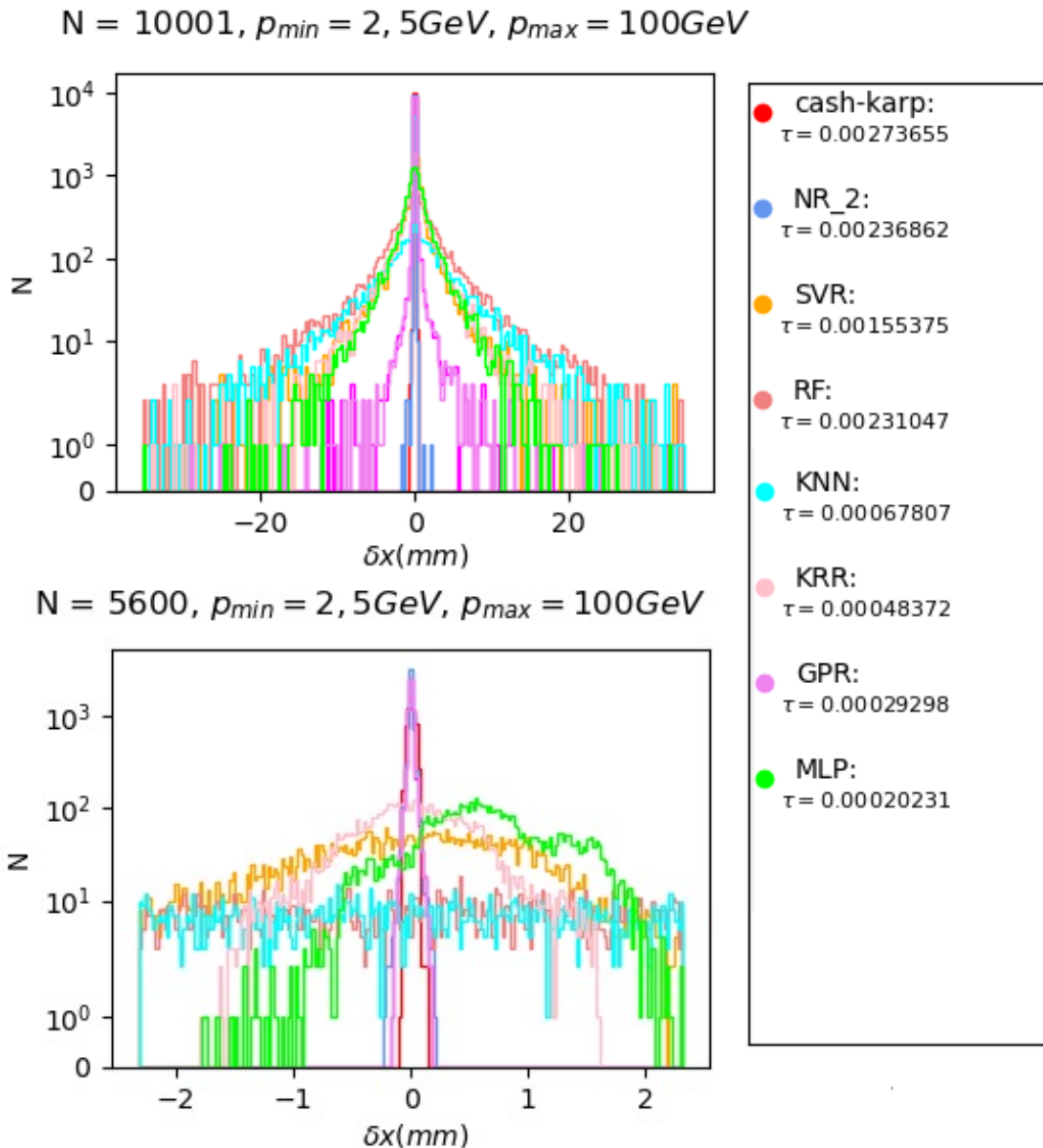


Slika 5.8: Na levi sliki so prikazane vrednosti parametrov SVD ekstrapolatorja pri razvoju z navadnimi polinomskimi členi, medtem ko so na desni prikazane vrednosti parametrov pri razvoju s Čebiševimi polinomi. V zgornjih vrsticah so prikazane vrednosti parametrov v linearni skali in v spodnji vrstici v logaritemski skali. Parametri so obarvani v štirih različnih barvah, eno za vsako od komponent izhodnega vektorja stanja (x, y, t_x, t_y) . Čebiševi polinomi preferirajo višje vrednosti pri parametrih, ki pripadajo členom nižjih redov.

5.3 Regresija z Gaussovimi procesi

S pristopom prilagajanja prostih parametrov nastavka več-dimenzionalne funkcije, na podlagi nekega učnega vzorca, se zelo približamo pravim metodam strojnega učenja. Takoj se pojavi vprašanje, ali so klasične metode strojnega učenja uporabne za hitro in natančno napoved sledi delcev, in če so, katere uporabiti. Tukaj se pogosto humorno navaja teorem brezplačnega kosila, s čimer želimo opozoriti, da ne moremo brez predhodnega znanja in testiranja intuitivno uganiti katera metoda bo delovala bolje od druge. Prav iz tega razloga smo izvedli preprosto študijo, kjer smo želeli uporabiti velik nabor klasičnih metod strojnega učenja in poskusiti oceniti, katera bi bila najbolj uporabna za dani problem vzvratne ekstrapolacije.

Pri tej študiji smo uporabili Python-ski paket `scikit-learn`[43], sama študija pa je namenjena kot površna analiza palete metod brez izostrovanja hiperparametrov. Kljub temu je bil cilj poiskati ne samo natančno ampak tudi hitro metodo. Rezultate študije lahko vidimo na sliki 5.9. Metode so se načeloma naučile reproducirati nekatere vzorce trajektorij, zlasti v primeru lažjega šopkastega podatkovnega seta, kjer so se metode zmotile za največ 4 mm pri ekstrapolaciji čez 5.000 mm. V primeru podatkovnega seta z minimalno pristranskostjo pa je bila natančnost znatno slabša. Na podlagi te začetne analize se je kot najbolj učinkovita izkazala metoda GPR - regresija z Gaussovimi procesi (ang. *Gaussian Process Regression*),



Slika 5.9: Študija klasičnih metod strojnega učenja za vzvratno ekstrapolacijo. Zgoraj je prikazana študija na podatkovnem setu z minimalno pristranskostjo, spodaj pa študija na šopkastem podatkovnem setu. Analiza je bila narejena na podlagi metod strojnega učenja iz paketa `scikit-learn`[43]. Na desni je časovno testiranje metod (v sekundah) za napoved 10.000 trajektorij iz podatkovnega seta z minimalno pristranskostjo. Uporabljene metode v študiji so bile; SVR - regresija s pomožnimi vektorji, RF - naključna drevesa, KNN - metoda najbližjih sosedov, KRR - regresija robov z jedri, GPR - regresija z Gaussovimi procesi in MLP - večplastna gosto povezana nevronska mreža[43].

saj je bila edina, ki je bila zmožna približno reproducirati natančnost že znanih ekstrapolatorskih funkcij. Druga najbližja metoda je bila gosto povezana nevronska mreža s petimi skritimi plastmi dimenzij (50, 100, 100, 100, 50), vendar pa je tudi po treniranju na celotnem podatkovnem setu imela sistematično pristranskost, ki je vidna na sliki 5.9. V nasportju z nevronske mreže je ironično metoda GPR tista,

ki je za napoved porabila najmanjši vzorec iz podatkovnih setov[43]. V naslednjem podpoglavju bomo najprej opisali matematično ozadje metode GPR nato pa njeno implementacijo.

5.3.1 Matematično ozadje

Izjemno uporabna pedagoška razlaga metode GPR leži v zgodovini njenega nastanka. Metoda je pogosto znana pod imenom *kriging* oz. napoved Wiener - Kolmogorova. Poimenovana je po južno-afriškem statistiku in rudarskemu inženirju Danie G. Krige-ju, ki je v sredini prejšnjega stoletja skušal oceniti porazdelitev zlata v zemlji na podlagi redkega števila izvrtin. Želel je modelirati korelacijo med posameznimi meritvami, na podlagi katerih bi nato lahko napovedal najverjetnejšo vrednost med njimi. Gaussova oz. normalna porazdelitev naključne spremenljivke nam je vsekakor domač koncept. Ko pa začnemo govoriti o enem izidu množice meritev, pa začnemo govoriti o *Gaussovem procesu*, ki ga opišemo z več-dimenzionalno normalno porazdelitvijo.

Pri regresiji z Gaussovimi procesi se ukvarjamo z izmerjeno množico podatkov \mathcal{D} iz n meritev, $\mathcal{D} = \{(\vec{x}_i, y_i) | i = 1, \dots, n\}$, kjer \vec{x} označuje vhodni vektor dimenzije D in y označuje skalarno vrednost izvida meritve. Predpostavimo, da so naše meritve y_i porazdeljene normalno okoli resničnih vrednostih preko Gaussovega šuma:

$$y = y_{res.} + \epsilon = f(\vec{x}) + \epsilon; \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2) \quad (5.14)$$

in želimo modelirati prave vrednosti $y_{res.}$. V splošnem lahko izberemo poljubno funkcijo za modeliranje $y_{res.}$, seveda je linearna najpreprostejša. Tukaj lahko dodamo še en nivo kompleksnosti in uvedemo funkcijo $\vec{\phi}(\vec{x})$, ki preslikava vhodni D -dimenzionalni vektor \vec{x} v N dimenzionalni prostor značilk (ang. *feature space*), in predstavlja bazo naših modelskih funkcij. Vendar v tej točki še ne rabimo razlikovati med njima in se lahko preprosto pretvarjamo, da lahko z $\vec{\phi}$ predstavljamo tako vhodni vektor \vec{x} ali pa nelinearno transformiran vektor, ki ga izračunamo z množico modelskih baznih funkcij. Sedaj lahko naš regresijski problem zapišemo kot:

$$y = \vec{\phi}(\vec{x})^T \vec{w} + \epsilon = \vec{\phi}^T \vec{w} + \epsilon \quad (5.15)$$

kjer si zopet lahko za potrebe preglednosti predstavljamo, da je $\vec{\phi}$ linearna funkcija \vec{x} (implicitno smo pretransformirali tudi šum $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$).

Najprej pomislimo, kakšna je verjetnostna porazdelitev naših meritev glede na parametre (ang. *likelihood*), faktorizirana čez vse meritve v podatkovnem setu (kjer predpostavimo, da so neodvisne), zaradi česar dobimo izraz

$$\begin{aligned} p(\{\Phi, \vec{y}\} | \vec{w}) &= \prod_{i=1}^n p(\{\vec{\phi}_i, y_i\} | \vec{w}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{(y_i - \vec{\phi}_i^T \vec{w})^2}{2\sigma_n^2}\right) \\ &= \frac{1}{(\sqrt{2\pi}\sigma_n)^{n/2}} \exp\left(-\frac{1}{2\sigma_n^2} |\vec{y} - \Phi^T \vec{w}|^2\right) = \mathcal{N}(\Phi^T \vec{w}, \sigma_n^2 I), \end{aligned}$$

ki efektivno opiše verjetnost, da smo izmerili vrednosti naših meritev glede na izbrani model. V smislu **Bayesovega pravila** hočemo izračunati posteriorno oz. naknadno porazdelitev uteži $p(\vec{w} | \{\Phi, \vec{y}\})$ preko inference izmerjenih meritev

$$\text{naknadna} = \frac{\text{verjetnost} \times \text{predhodna}}{\text{marginalna verjetnost}}, \quad p(\vec{w} | \{\Phi, \vec{y}\}) = \frac{p(\{\Phi, \vec{y}\} | \vec{w})p(\vec{w})}{p(\{\Phi, \vec{y}\})}, \quad (5.16)$$

kjer je normalizacijska konstanta, znana tudi pod imenom marginalna verjetnost (ang. *marginal likelihood*), neodvisna od uteži in podana preko formule

$$p(\{\Phi, \vec{y}\}) = \int p(\{\Phi, \vec{y}\}|\vec{w})p(\vec{w})d\vec{w}. \quad (5.17)$$

Posteriorna verjetnostna porazdelitev združi verjetnost in predhodno verjetnost (ang. *prior*), in s tem zajame vse, kar vemo o parametrih. Izbira predhodne porazdelitve je večinoma stvar ugibanja, ki včasih meji na neke vrste čarobno intuicijo, kljub temu bomo zaradi preprostosti predpostavili za predhodno porazdelitev zgolj še eno normalno porazdelitev $\vec{w} \sim \mathcal{N}(\vec{0}, \Sigma_p)$, ki bo v bistvu opisala naše pomankljivo znanje o optimalni izbiri uteži. Če zapišemo zgolj člene iz verjetnosti in predhodne porazdelitve, dobimo naslednji izraz

$$\begin{aligned} p(\vec{w}|\{\Phi, \vec{y}\}) &\propto \exp\left(-\frac{1}{2\sigma_n^2}(\vec{y} - \Phi^T \vec{w})^T(\vec{y} - \Phi^T \vec{w})\right) \exp\left(-\frac{1}{2}\vec{w}^T \Sigma_p^{-1} \vec{w}\right) \\ &\propto \exp\left(-\frac{1}{2}(\vec{w} - \vec{w}')^T \left(\frac{1}{\sigma_n^2} \Phi \Phi^T + \Sigma_p^{-1}\right) (\vec{w} - \vec{w}')\right) \\ &\sim \mathcal{N}(\vec{w}' = \frac{1}{\sigma_n^2} A^{-1} \Phi \vec{y}, A^{-1}), \quad A = \frac{1}{\sigma_n^2} \Phi \Phi^T + \Sigma_p^{-1}, \end{aligned}$$

Tukaj opozorimo, da specifično za ta model (in v resnici za vsako Gaussovo posteriorno porazdelitev) dobimo, da je povprečna vrednost posteriorne porazdelitve $p(\vec{w}|\{\Phi, \vec{y}\})$ tudi enaka njeni maksimalni vrednosti², kar ang. imenujemo tudi **maximum a posteriori (MAP)** ocena optimalnih vrednosti \vec{w} . Pri MAP oceni, uporabimo največjo vrednost posteriorne porazdelitve, da preko nje izpeljemo oceno katerekoli spremenljivke, ki nas zanima. Maksimalno vrednost porazdelitve lahko definiramo zgolj pri porazdelitvah z enim vrhom (ang. *unimodal*). Tukaj pa opozorimo, da maksimalna vrednost porazdelitve ne sovpada vedno z njeno povprečno vrednostjo. Ta razlika je pomembna pri asimetričnih porazdelitvah[44].

Napoved na novem, še neizmerjenem vhodnem vektorju, dobimo tako, da povprečimo preko vseh možnih vrednosti parametrov, ki pa jih utežimo z njihovo posteriorno porazdelitvijo. Porazdelitev napovedi je zopet normalna s povprečno vrednostjo, ki je podana preko povprečja posteriorne porazdelitve uteži pomnožene s testnim vhodnim vektorjem[44]:

$$p(\{\vec{\phi}_*, f_*\}|\{\Phi, \vec{y}\}) = \int p(\{\vec{\phi}_*, f_*\}|\vec{w})p(\vec{w}|\{\Phi, \vec{y}\})d\vec{w} = \mathcal{N}\left(\frac{1}{\sigma_n^2} \vec{\phi}_*^T A^{-1} \Phi \vec{y}, \vec{\phi}_*^T A^{-1} \vec{\phi}_*\right). \quad (5.18)$$

5.3.2 Implementacija algoritma

Sedaj, ko smo spoznali matematično ozadje metode GPR, se lahko zelo hitro ustrašimo abstraktne statistike, na podlagi katere deluje, brez da bi razumeli, kako praktično implementirati tak algoritem. Izpeljali smo več verjetnostnih porazdelitev, ki jih še enkrat naštejmo. Marginalna verjetnost je neodvisna od uteži in služi zgolj kot normalizacijska konstanta, zaradi česar jo v praktični implementaciji algoritma preprosto izpustimo. Predhodna verjetnost je rahlo misteriozna, ampak kot smo videli

²Maksimalno vrednost porazdelitve tipično v literaturi najdemo pod imenom ang. *mode*.

je zgolj definirana s pomočjo Σ_p , ki ga bomo omenili kasneje. Posteriorna porazdelitev oz. MAP je najpomembnejši del algoritma, ker je prilagajanje našega modela ekvivalentno maksimizaciji logaritma verjetnosti $\mathcal{L} = p(\vec{w}|\{\Phi, \vec{y}\})$ (logaritmiramo, ker se želimo izogniti numeričnim napakam zaokroževanja).

$$\mathcal{L} = \frac{1}{\sqrt{(2\pi)^n \det(A^{-1})}} \exp\left(-\frac{1}{2}(\vec{w} - \vec{w}')^T A(\vec{w} - \vec{w}')\right) \quad (5.19)$$

$$-\ln(\mathcal{L}) = \ln(\sqrt{(2\pi)^n}) + \ln(\sqrt{\det(A^{-1})}) + \frac{1}{2}(\vec{w} - \vec{w}')^T A(\vec{w} - \vec{w}') \quad (5.20)$$

Naš problem prilagajanja se tako spremeni v minimizacijski problem. Ker pa imamo opravka z normalnimi porazdelitvami, lahko analitično izračunamo odvode cenovne funkcije in tudi globalni minimum za uteži $\partial \ln(\mathcal{L})/\partial w_i = 0$, kar nam da za optimalne vrednosti uteži že poznane rezultate

$$\vec{w} = \frac{1}{\sigma_n^2} \left(\frac{1}{\sigma_n^2} \Phi^T \Phi + \Sigma_p^{-1} \right)^{-1} \Phi^T \vec{y}, \quad (5.21)$$

Podobno lahko analitično izpeljemo tudi minimum za kovariančno matriko po zvezi $\partial \ln(\mathcal{L})/\partial \sigma_n^2 = 0$. Na koncu vse te izraze znova vstavimo v našo posteriorno verjetnost in pridobimo maksimalno vrednost, s čimer smo dobili popolnoma neparametrično metodo. Tukaj pa pridejo v igro naše modelirske funkcije $\vec{\phi}(\vec{x})$, ki so običajno odvisne od majhnega števila hiperparametrov. Optimalno vrednost teh hiperparametrov $\vec{\theta}$ moramo pridobiti na podlagi numerične minimizacije. Velika prednost modela je tudi tukaj, saj lahko analitično izračunamo vrednosti gradientov naše cenovne funkcije, zaradi česar se lahko pri numerični minimizaciji zatečemo k metodam gradientnega spusta kot je optimizator Adam [43].

V tej točki uvedemo še dodaten trik, ki drastično izboljša ta model. Imenuje se jedrni trik oz. ang. **kernel trick**[44] in ga opišemo s pomočjo jedrne funkcije $k(\vec{x}, \vec{x}') = \phi(\vec{x})^T \Sigma_p \phi(\vec{x}') = \phi'(\vec{x})^T \phi'(\vec{x}')$ (celo naša predhodna kovarianca oz. Bayesov šum se vključi kot prost parameter)³. Jedro je neke vrste kovariančna funkcija, ki je izjemno pomembna, saj z njeno uvedbo lahko iz definicije problema izločimo kakršnokoli eksplicitno odvisnost od baznih modelskih funkcij $\vec{\phi}$. To predstavlja izjemno prednost, ker nam omogoča konfiguracije modela, kjer našo prilagoditveno modelsko funkcijo implicitno razvijemo z neskončno vrsto nelinearnih baznih funkcij $\vec{\phi}$, tako da jih pospravimo znotraj funkcije jedra (najbolj poznano jedro je znano pod imenom RBF - radialnih baznih funkcij[43]). Na koncu se naša napoved prepíše na podlagi jedrne funkcije v obliko

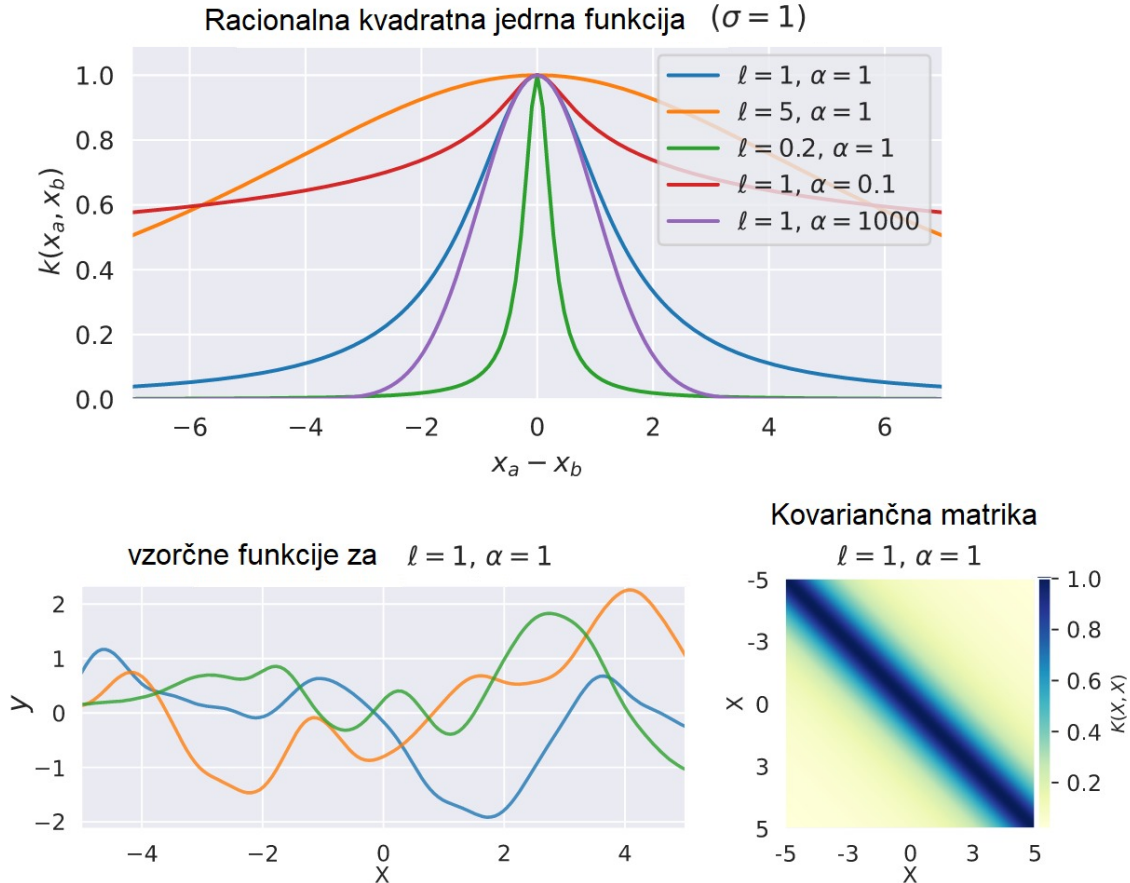
$$\vec{y}_* = K(X_*, X)^T (K(X, X) + \sigma_n^2 I)^{-1} \vec{y} \quad (5.22)$$

$$\sigma_{\vec{y}_*}^2 = K(X_*, X_*) + K(X_*, X)^T (K(X, X) + \sigma_n^2 I)^{-1} K(X, X_*) \quad (5.23)$$

Tukaj moramo biti pozorni na notacijo vektorjev in matrik. Ker je vhodni vektor \vec{x}_i lahko D -dimenzionalen, smo uvedli notacijo matrik, s katerim opišemo vektor vektorjev. Jedrna funkcija je tipično definirana s parom dveh vhodnih vektorjev, ki sta lahko več-dimenzionalna. V takšnem modelu eksplicitno ne izbiramo modelskih funkcij na vhodnem vektorskem prostoru marveč poskušamo modelirati korelacijske

³Ker je Σ_p pozitivno definitna lahko definiramo njen koren preko metode SVD.

funkcije med našimi meritvami. Pri tem potrebujemo popolnoma drugačno modelsko intuicijo in čeprav zveni zasnovanje novih jedrnih funkcij kot izjemno kompleksna in abstraktna naloga, nam **Mercerjev teorem**[44] definira kritična pravila za njihovo konstrukcijo. V najbolj preprostih primerih lahko že zasnovane jedrne funkcije združujemo skupaj v nove na podlagi nekaterih osnovnih matematičnih operacij[45].



Slika 5.10: Zgoraj je prikazana oblika racionalne kvadratne jedrne funkcije pri različnih vrednosti parametrov. Jedrne funkcije poskušajo opisati korelacijo med meritvami, zato tipično njihova vrednost pada z oddaljenostjo med meritvama. Spodaj pa so prikazane tipične modelske funkcije, ki jih dobimo z vzorčenjem predhodne porazdelitve uteži. Zraven na desni je prikazana kovariančna matrika jedrne funkcije `scikit-learn`[43].

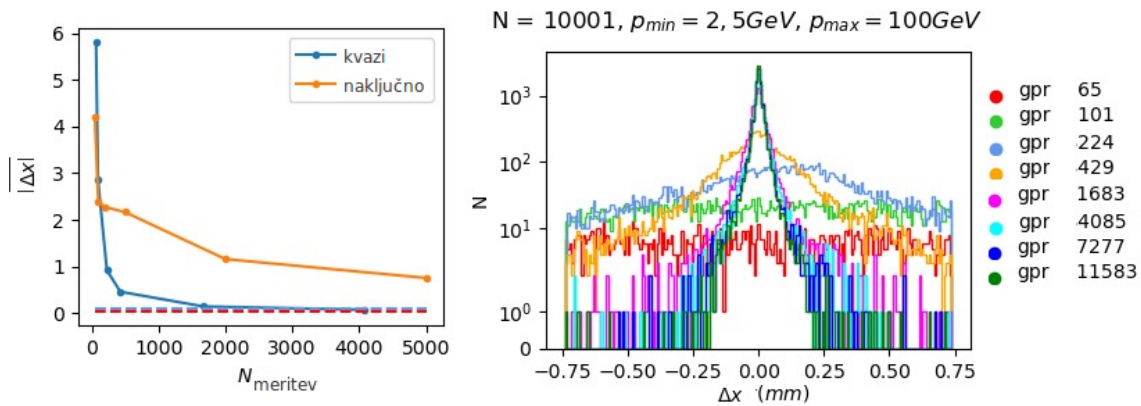
Pri implementaciji metode GPR za ekstrapolacije v vzvratno smer čez magnet smo za jedrno funkcijo izbrali t.i. racionalno kvadratno funkcijo. Ta je po obliki zelo podobna RBF jedrni funkciji in je podana po formuli:

$$k(\vec{x}_a, \vec{x}_b) = \sigma^2 \left(1 + \frac{\|\vec{x}_a - \vec{x}_b\|^2}{2\alpha l^2} \right)^{-\alpha}. \quad (5.24)$$

Vendar smo fiksirali tako amplitudo $\sigma^2 = 1$ kot tudi potenco $\alpha = 1$. Z oznako l smo označili skalirno dolžino, ki v jedrni funkciji določa vztrajnost koleracij meritev glede na njihovo oddaljenost. Gre v resnici za Taylorjev razvoj RBF jedrne funkcije, s čimer se izognemo počasnemu računanju eksponentnih funkcij. S fiksiranjem potence pa omogočimo hitrejšo računanje odvodov jedrnih funkcij, ki jih potrebujemo

Poglavje 5. Nove metode za hitro ekstrapolacijo trajektorij

tudi za analitičen izračun Jacobijeve matrike. Vizualizacija racionalne kvadratne jedrne funkcije je prikazana na sliki 5.10. Poleg dveh parametrov jedrne funkcije metoda GPR ne vsebuje nobenih preostalih koeficientov, saj je model direktno prilagojen na podane meritve. Natančnost metode tako izboljšujemo in izostrujemo, podobno kot pri SVD ekstrapolatorju, preko učnega seta podatkov. Medtem ko je sam proces prilagajanja računsko bolj zahteven v metodi GPR, pa nam ta omogoča hitro konvergenco že pri majhnih velikostih učnega seta podatkov, kot lahko vidimo na sliki 5.11.



Slika 5.11: Na levem grafu vidimo konvergenco širine porazdelitve odstopanj napovedane komponente končnega vektorja stanja z metodo GPR pri večanju učnega vzorca podatkovnega seta z minimalno pristranskostjo. Na desni so prikazane te porazdelitve GPR modelov, ki so bili naučeni na učnih vzorcih pridobljeni s kvazi-naključnim vzorčenjem. Optimalno velikost učnega vzorca dosežemo pri 4.000 trajektorijah. Na levem grafu sta s črtkanima črtama označeni povprečna napaka RK metode (z rdečo) in NR sheme parametriziranega ekstrapolatorja (z modro) `scikit-learn`[43].

Kot pomemben detajl je potrebno omeniti, da zgoraj izpeljana matematična definicija GPR modela drži zgolj za eno-dimenzionalne skalarne vrednosti meritev. V primeru naše ekstrapolacije pa imamo opravka z izhodnim vektorjem s štirimi komponentami. Take probleme najpreprosteje rešujemo s štirimi neodvisnimi GPR modeli, eden za vsako komponento izhodnega vektorja. Potencialna nadgraditev v takih problemih je znana pod imenom več-dimenzionalnih GPR modelov⁴. Te nadgrajeni modeli poskušajo modelirati korelacije med indeksi komponent izhodnega vektorja, kar tipično zahteva uvedbo sekundarne jedrne funkcije na diskretnem prostoru indeksov izhodnega vektorja. Takšna nadgraditev pa ne doprinese vedno k izboljšanju metode, hkrati pa bistveno oteži časovno zahtevnost napovedovanja, saj učinkovito umetno poveča učni nabor meritev[46][47].

Na podlagi rezultatov časovne analize v tabeli 5.2, ki je bila izvedena v Python-skem peskovniku LHCb implementacije RK algoritma, lahko vidimo, da so ekstrapolacije v vzvratno smer znatno bolj zahtevne in optimalna izbira metod ni trivialna. Časovno zahtevnost NR sheme je pričakovana glede na časovne analize ekstrapolacije naprej iz tabele 5.1. Smer ekstrapolacije prav tako ne vpliva bistveno na RK algoritem. Kot se izkaže je metoda GPR relativno počasnejša, razlog za to je preprosto količina koeficientov. Metoda GPR deluje hitro v primeru majhnih učnih setov.

⁴V literaturi so številna poimenovanja kot na primer MOGP - ang. *multioutput* GPR, ali pa MTGP - ang. *multitask* GPR.

Tabela 5.2: Časovno testiranje ekstrapolatorjev v vzvratni smeri skozi magnet. Časovna analiza je bila izvedena v Python-skem peskovniku, v katerem smo simulirali delovanje LHCb implemetnacije adaptivne RK Cash-Karp metode. Časovno testiranje je bilo izvedeno pri napovedovanju 10.000 trajektorij iz podatkovnega seta z minimalno pristranskostjo. Za testiranje metode GPR je bil uporabljen Python-ski paket `scikit-learn`[43].

Metoda	RK Cash-Karp	NR-LIN	NR-KVAD	SVD	GPR
Čas (ns)	265.565	156.346	227.367	37.389	591.293

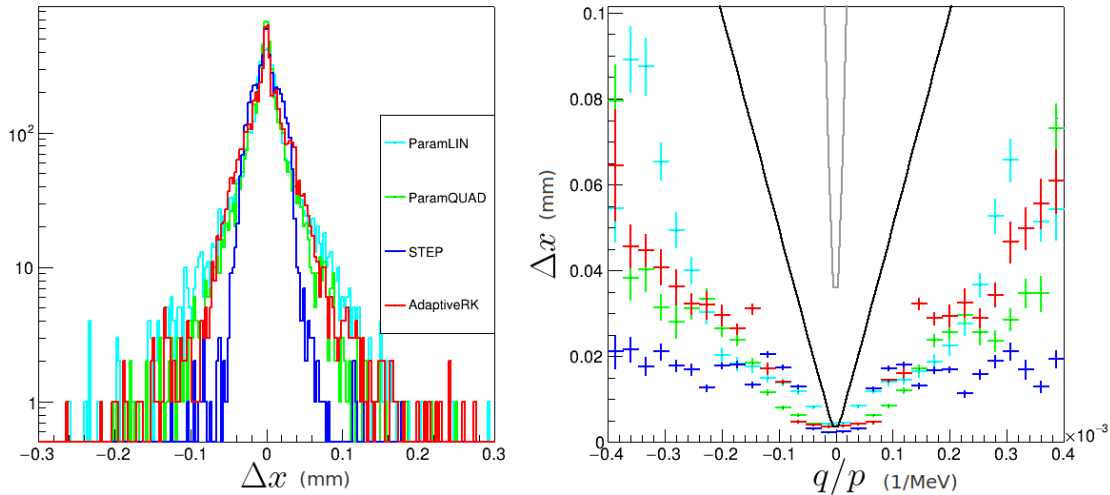
Da smo lahko zadovoljili potrebni natančnosti, smo morali za metodo GPR vključiti 4.000 trajektorij iz podatkovnega seta z minimalno pristranskostjo. Metoda bi delovala hitreje, če bi jo naučili na manjšem številu trajektorij. Izboljšavo bi morda lahko našli v načinu vzorčenja podatkovnega seta. Pokazali smo, da naključno vzorčenje podatkovnih setov ne vodi k dovolj hitri konvergenci in s tem dovolj dobri natančnosti metode. Prav tako smo pokazali, da je kvazi-naključno vzorčenje bistveno pomagalo pri postopku učenja. Potencialna izboljšava v prihodnosti bi bila poskusiti najti ravnovesje med simuliranimi kvazi-naključnimi podatkovnimi seti in urejenimi kubičnimi mrežami (na kateri temelji parametrizirani ekstrapolator), za razvoj novih podatkovnih setov, ki bi izboljšali konvergenco metode GPR. Dodatna motivacija za razvoj hiperkubičnega podatkovnega seta trajektorij je, da je tipično možno pohitriti računanje jedrnih funkcij v metodi GPR, ko so učni podatki urejeni na več-dimenzionalni urejeni mreži. Kot najhitrejša metoda se je izkazal nastavek, ki ga določimo z metodo SVD. Razlog najverjetneje leži v Python-skem paketu `numpy`[48], saj je nastavek iz enačbe 5.9 preprost za računanje napovedi z majhnim številom parametrov. Medtem ko lahko komponente Jacobijeve matrike analitično izračunamo za vse metode, je slabost metode SVD v tem, da nimamo nobenega načina, preko katerega bi lahko ocenili, ali lahko zaupamo napovedi. Ravno to je največja prednost metode GPR, saj nam ta pravilno oceni tudi napako napovedi. Tukaj je zopet potrebno opomniti, da je, z izjemo algoritma RK, teoretično možno preostale ekstrapolatorje paralelizirati, s čimer bi lahko ekstrapolatorji propagirali več trajektorij hkrati.

5.4 Rezultati

Medtem ko smo metode originalno zasnovali na Python-skem peskovniku implemetnacije LHCb estrapolatorskega programskega orodja, je bil cilj naloge seveda implementirati metode v dejanskem rekonstrukcijskem programu znotraj programske arhitekture `Gaudi` in tudi tam replicirati predvideno natančnost metod. Osredotočili smo se na ekstrapolacije skozi magnet, saj so te ekstrapoalcije najbolj zahtevne zaradi najdaljše dolžine ekstrapolacij in močnega neenakomernega magnetnega polja.

5.4.1 Ekstrapolacije naprej

Pri ekstrapolaciji naprej čez magnet deluje parametrizirani ekstrapolator, ki ga je predlagal prof. Pierre Billoir[22], vsekakor izjemno učinkovito in hitro. Skupaj



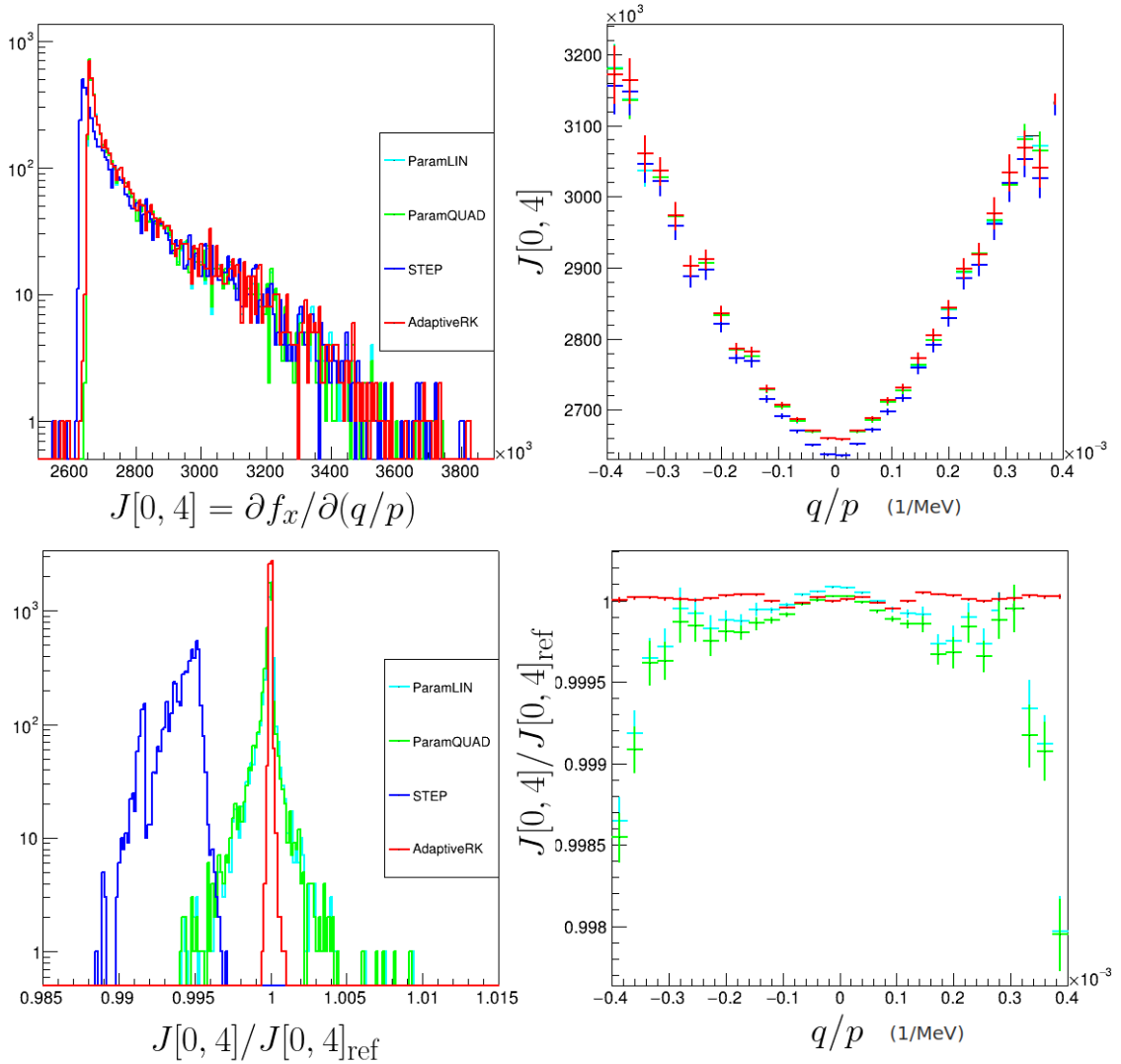
Slika 5.12: Natančnost parametriziranega ekstrapolatorja pri ekstrapolacijah naprejškozi magnet. Z rdečo je označen RK ekstrapolator, z modro pospešena verzija RK algoritma z imenom STEP ekstrapolator in z zeleno ter ciano parametrizirani ekstrapolator. S sivo in črno črto je označena ocenjena zunanja napaka oz. ena desetina te vrednosti. Rezultati so bili ocenjeni na podatkovnem setu z minimalno pristranskostjo, za referenco pa je bil uporabljen RK algoritem s fiksnim korakom dolžine 10 mm[39][22].

z bikvadratno interpolacijo parametriziranih koeficientov je sposoben ekstrapolirati trajektorije z enako natančnostjo kot standarden RK ekstrapolator, kar očitno lahko vidimo na sliki 5.12.

Skupaj s parametriziranim ekstrapolatorjem smo primerjali tudi delovanje t.i. STEP ekstrapolatorja[21], pri katerem gre za poskus vektorizirane in pospešene verzije standardnega adaptivnega RK ekstrapolatorja. Ta je sposoben izračunati natančne in hitre napovedi vektorjev stanja vendar pa ne oceni pravilno komponent Jacobijeve matrike. Analiza natančnosti komponent Jacobijeve matrike ni najbolj enostavna predvsem zaradi njegove bolj abstraktne vloge v Kalmanovem filtru, kjer v koraku napovedi 3.10 propagira kovariančno matriko povezano z meritvijo vektorja stanja. Na sliki 5.13 lahko vidimo porazdelitev napovedane komponente Jacobijeve matrike $J_{04} = df_x/d(q/p)$, ki je ena izmed bolj ključnih komponent. Napačne vrednosti komponent Jacobijeve matrike povzročijo napačno propagacijo kovariančne matrike, s čimer pridobimo sistematično napako v končni napovedi Kalmanovega filtra. Komponenta J_{04} je posebno pomembna saj ima direkten vpliv na resolucijo gibalne količine. Dobra resolucija gibalne količine je ključna za dobro masno resolucijo, še posebej pri dvo-delčnih razpadih, kjer potrebujemo visoko občutljivost pri iskanju redkih razpadov. Medtem ko na sliki 5.13 vidimo, da se pri napovedi komponente J_{04} zmoti tudi parametrizirani ekstrapolator, ta nima sistematičnega odstopanja kot STEP ekstrapolator. Prav tako so odstopanja večja za manj pomembne trajektorije z nizkimi gibalnimi količinami.

Za konec naj omenim, da smo v preteklosti že preučevali in analizirali natančnost parametriziranega ekstrapolatorja. Izkazal se je za izjemno dobrega kandidata za hitre ekstrapolacije, ki ga lahko v prihodnosti dodatno pospešimo z vektorizacijo. Ekstrapolacije s parametriziranim ekstrapolatorjem smo tako preučevali pri ekstrapolacijah čez magnet in tudi med prvima dvema detektorjema, torej VELO,

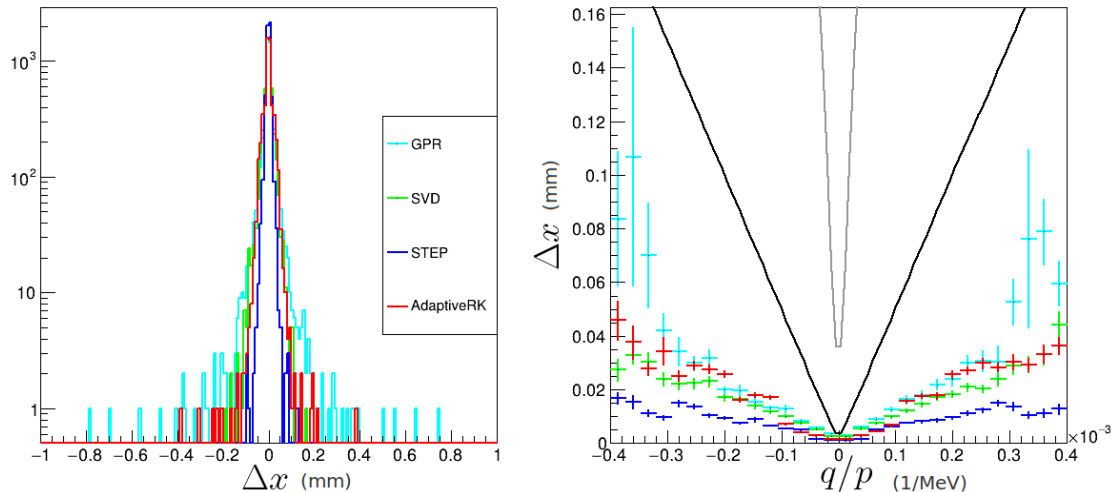
in sledilnim sistemom pred magnetom. Parametrizirani ekstrapolator je bil v tem režimu še posebno natančen, prav tako pa smo ga lahko v tem režimu uporabili za vzvratno ekstrapolacijo brez uporabe NR iteracijske sheme, saj pred magnetom trajektorije še ne občutijo znatnega ukrivljanja. Rezultati teh testiranj s parametriziranim ekstrapolatorju so v dodatku D [39][22].



Slika 5.13: Na zgornjih grafih so prikazane porazdelitve napovedanih J_{04} komponent za različne ekstrapolatorje. V spodnjih dveh vrsticah pa so prikazane porazdelitve razmerij napovedanih vrednosti z referenčno vrednostjo. Zlasti iz spodnjih grafov je očitna sistematična napaka izračunane komponente Jacobijeve matrike STEP ekstrapolatorja[39][22][21].

5.4.2 Ekstrapolacije nazaj

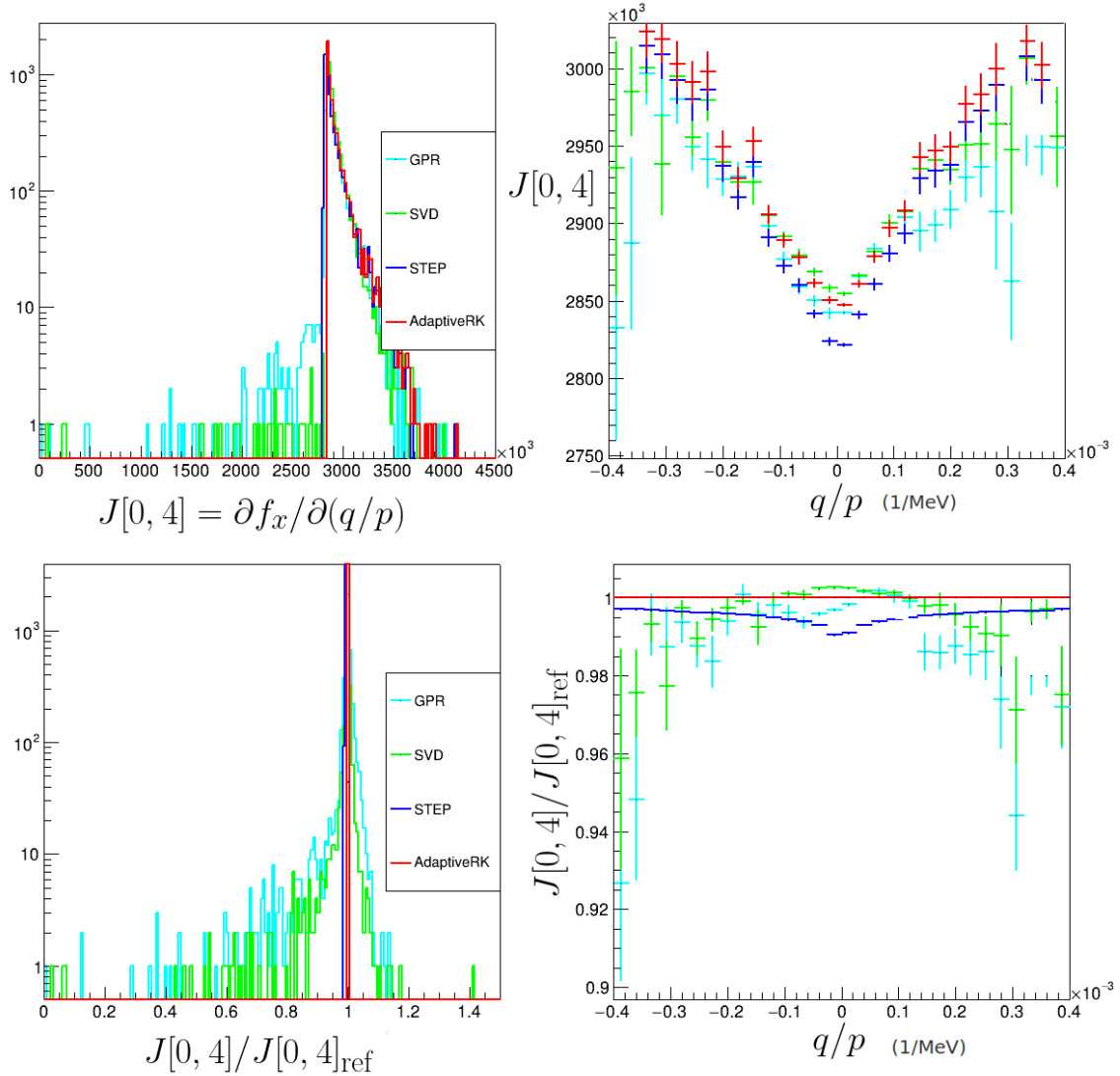
Pri ekstrapolacijah nazaj skozi magnet s trenutno programsko arhitekturo nismo uspeli implementirati in potestirati NR iterativne sheme parametriziranega ekstrapolatorja, ki bi delovala na podlagi izostrovanja začetne aproksimacije trajektorije iz prožilnega sistema HLT. Tako smo se osredotočili na testiranje alternativnih pristopov direktnega napovedovanja končnih vektorjev stanja trajektorije. Pri konč-



Slika 5.14: Natančnost SVD in GPR ekstrapolatorja pri ekstrapolacijah nazaj skozi magnet. Z rdečo je označen RK ekstrapolator, z modro pospešena verzija RK algoritma z imenom STEP ekstrapolator, z zeleno je označen SVD ekstrapolator in s ciano GPR ekstrapolator. S sivo in črno črto je označena ocenjena zunanja napaka in oz. ena desetina te vrednosti. Rezultati so bili ocenjeni na podatkovnem setu z minimalno pristranskostjo, za referenco pa je bil uporabljen RK algoritem s fiksnim korakom dolžine 10 mm.

nem prilagajanju metod smo tako za SVD kot tudi za GPR ekstrapolator uporabili podatkovni set, ki je vseboval tako trajektorije iz podatkovnega seta z minimalno pristranskostjo kot tudi trajektorije iz šopkastega podatkovnega seta. SVD ekstrapolator je vseboval nastavek, ki je vseboval produkte Čebiševih polinom do petega reda za komponente vhodnega vektorja q/p , x in y ter četrtega reda za preostali dve komponenti t_x in t_y . Skupno je vseboval 2.000 koeficientov, ki so bili prilagojeni na podlagi 50.000 trajektorij vzorčenih s kvazi-naključnim vzorčenjem. Končna verzija GPR ekstrapolatorja je bila prilagojena na 4.000 trajektorijah, zopet vzorčenih s kvazi-naključnim vzorčenjem. Za jedrno funkcijo je bila uporabljena racionalna kvadratna funkcija, ki so si jo delile vse komponente izhodnega vektorja.

Na sliki 5.14 lahko vidimo, da sta oba ekstrapolatorja izkazala natančnost boljšo od ene desetine ocenjene zunanje napake, pri čimer je metoda GPR pokazala asimetrično porazdelitev odstopanj za trajektorije z najnižjimi gibalnimi količinami. Na sliki 5.15 pa lahko vidimo še porazdelitev komponente Jacobijeve matrike J_{04} . Obe ekstrapolacijski metodi pokažeta podobno obnašanje. Napoved odvodov iz Jacobijeve matrike so slabše od parametriziranega ekstrapolatorja, vendar tukaj izvajamo ekstrapolacijo nazaj, za katero parametriziranega ekstrapolatorja brez NR sheme sploh ne moremo uporabiti. Za trajektorije delcev z gibalno količino večjo od 5 GeV/c opišeta obe metodi odvod J_{04} bolje kot pa alternativni STEP ekstrapolator, medtem ko pa je napoved slaba za počasne trajektorije. Tukaj je potrebno omeniti, da ker smo izbrali 2,5 GeV/c kot skrajni rob v učnem naboru podatkov, je smiselno pričakovati neoptimalno napoved na omenjenem robu. S to analizo smo uspeli pokazati, da sta obe metodi GPR in SVD zelo splošni in lahko na njuno kvaliteto ekstrapolacij močno vplivamo na podlagi učnega nabora. Prav tako ju je, zaradi njune splošnosti, enostavno uporabiti za ekstrapolacije v detektorjih s kompleksnimi in netrivialnimi geometrijami. Pri napovedi komponent Jacobijeve matrike z metodo



Slika 5.15: Študija napovedi Jacobijeve matrike SVD in GPR ekstrapolatorja pri esktrapolacijah nazaj skozi magnet. Na zgornjih grafih so prikazane porazdelitve napovedanih J_{04} komponent za različne ekstrapolatorje. V spodnjih dveh vrsticah pa so prikazane porazdelitve razmerij napovedanih vrednosti z referenčno vrednostjo.

GPR je za nesimetrične porazdelitve odstopanj najverjetneje kriv način vzorčenja podatkovnega seta. Ker je skalirna dolžina l , iz jedrne funkcije 5.24, enaka povsod po faznem prostoru meritev, so večje luknje v faznem prostoru meritev izjemno problematične pri napovedi metode GPR na tem območju. Zato tipično metoda GPR deluje najboljše kadar so meritve približno enakomerno razporejene med seboj. Metoda GPR pri prilagajanju vključi res majhno število trajektorij, ki pa niso nujno enakomerno razporejene po faznem prostoru. Ta problem smo sicer v veliki meri odpravili z uvedbo kvazi-naključnega vzorčenja, vendar pa bi idealno želeli trajektorije urediti po nekakšni pet-dimenzionalni hiperkubični mreži, kar bi poleg izboljšane napovedi omogočilo tudi še hitrejšo računanje vrednosti jedrne funkcije.

Časovne analize novo implementiranih metod na polnem računalniškem okolju LHCb nismo mogli narediti. Trenutno implementirane nove metode namreč še niso optimizirane za arhitekturo LHCb programskega ogrodja, medtem ko je standardni ekstrapolator RK bil deležen številnih postopnih izboljšav, zaradi česar časovna ana-

liza ne bi bila reprezentativna. Kljub temu izpostavimo, da vsi novo implementirani ekstrapolatorji uspešno napovejo trajektorije pod zadanim pragom desetine zunanje napake 5.7. Parametrizirani ekstrapolator deluje izjemno natančno pri ekstrapolacijah naprej pred magnetom in vsebuje tudi zelo natančne izračune Jacobijeve matrike. Alternativna oblika za vzvratno ekstrapolacijo temelji na NR shemi in potrebuje vsaj dve ekstrapolaciji za zadovoljivo natančnost. Ekstrapolator metode SVD deluje hitro na podlagi majhnega števila koeficientov, vendar nam ne ponuja nobenega statističnega jamstva o natančnosti svoje napovede. Metoda GPR trenutno deluje najpočasneje od novo implementiranih metod in ima tudi največja odstopanja pri opisu Jacobijeve matrike, vendar pa ima ravno metoda GPR največ priložnosti za izboljšave v prihodnosti. Asimetrična odstopanja v Jacobijevi matriki so močan indikator o sistematični pristranskosti modela, zaradi česar sklepamo, da je potrebno za metodo GPR poiskati alternativni učni nabor trajektorij urejenih na več-dimenzionalni hiperkubični mreži, na podlagi katere bi lahko optimizirali tudi izračun vrednosti jedrnih funkcij[47][46][43].

Poglavje 6

Zaključek

V uvodu smo na kratko predstavili cilje in motivacije meritev na eksperimentu LHCb. V drugem poglavju smo se seznanili z obliko detektorja LHCb in predvsem z lastnostmi magnetnega polja, ki otežujejo ekstrapolacije trajektorij nabitih delcev skozi območje magneta. Ob tem smo se poglobili v kompleksno in neenakomerno prostorsko odvisnost magnetnega polja dipolnega magneta na LHCb. Izpostavili smo problem artefaktnih fluktuacij na robu sredice magneta, ki nam onemogoča iskanje preprostejših ekstrapolatorjev izven stožčaste akceptance detektorja LHCb. Z novim “Run 3” obratovanjem LHC s pet-kratno luminoznostjo žarkov[8] bo ogromna količina zajetih podatkov predstavljala znatno večje breme za rekonstrukcijsko programsko opremo. Kot najpočasnejši člen v programski arhitekturi smo izpostavili standardni RK ekstrapolator, ki temelji na adaptivni numerični integraciji in ga je zato zelo zahtevno računsko pospešiti v smislu vektorizacije.

Predstavili smo tri metode, ki omogočajo hitro napoved trajektorij delcev brez iterativnega korakanja. S formulacijo novih metod smo se izognili numerični integraciji enačb gibanj za nabite delce v magnetnem polju in s tem tudi branju vrednosti samega magnetnega polja. Parametrizirani ekstrapolator, ki ga je zasnoval prof. Pierre Billoir[22], temelji na perturbativnem razvoju okoli idealnih trajektorij, ki izvirajo točno iz izhodišča. Metoda omogoča prepis iz tri-dimenzionalnega zapisa vrednosti magnetnega polja in trilinearne interpolacije, v dvo-dimenzionalno ravninsko parametrizacijo s kvadratno interpolacijo. Metoda je hitra, vendar je zasnovana za ekstrapolacije v smeri vzdolž žarkovne osi. Ključnega pomena za prilagajanje trajektorij je uporaba Kalmanovega filtra, ki potrebuje ekstrapolacije v obratno smer. Parametrizirani ekstrapolator tako nadgradimo s počasnejšo dvokoračno Newton-Raphson shemo za vzvratne ekstrapolacije. Alternativno smo predstavili še dve metodi za ta tip ekstrapolacij, ki temeljita na učenju iz učnega nabora trajektorij. Pri metodi SVD privzamemo več-dimenzionalni funkcionalni nastavek, v katerem prosti parametri nastopajo zgolj linearno in ga zato lahko prilagodimo z minimizacijo χ^2 na učnem naboru trajektorij, pri čimer predločen sistem enačb nato rešimo s pomočjo SVD dekompozicije. Metoda se je izkazala za najhitrejšo in zelo natančno, vendar ne vsebuje nobene statistične ocene o natančnosti svoje napovedi. Kot zadnjo metodo smo predstavili pravo metodo strojnega učenja, metodo GPR, ki na podlagi majhnega nabora trajektorij modelira korelacije med meritvami, na podlagi katerih izračuna verjetnostne porazdelitve okoli točk, v katerih nismo podali meritev. Metoda GPR se je izkazala za najpočasnejšo izmed metod, zaradi prevelikega števila prilagoditvenih trajektorij, vendar pa ponuja statistično oceno o lastni napovedi.

Pokazali smo, da pri učenju metod strojnega učenja za rekonstrukcijo trajektorij igra ključno vlogo uravnotežen in reprezentativen učni nabor trajektorij. Pri našem problemu smo to dosegli s pomočjo kvazi-naključnega vzorčenja trajektorij, v prihodnosti pa vidimo izboljšavo v zasnovi več-dimenzionalnega hiperkubičnega podatkovnega seta, katerega mrežasto strukturo bi lahko uporabili za hitrejše računanje jedrnih funkcij metode GPR. V Python-skem peskovniku smo demonstrirali časovno prednost parametriziranega ekstrapolatorja in metode SVD v primerjavi s standardnim RK algoritmom. Dodatno teoretično pohitritev lahko dobimo pri računanju Jacobijeve matrike, ki jo morajo izračunati ekstrapolacijske funkcije. Velika prednost novih metod je, da izračunajo Jacobijevo matriko analitično. Poleg tega ima funkcionalna oblika novih metod teoretično prednost pri implementaciji algoritmov na moderne računalniške arhitekture, preko katerih bi bilo možno ekstrapolirati skupine trajektorij hkrati, česar algoritem RK ne zmore. Parametrizirani ekstrapolator je dokončno zasnovan in pripravljen na realistične rekonstrukcije trajektorij saj omogoča tudi izjemno natančen izračun komponent Jacobijeve matrike. Metodi SVD in GPR pa potrebujeta dodatno študijo asimetričnih odstopanj napovedi komponent Jacobijeve matrike, ki se jo potencialno da odpraviti preko boljšega vzorčenja podatkovnega nabora trajektorij.

Literatura

- [1] E. Bos, *Reconstruction of charged particles in the LHCb experiment*, Doktorska disertacija, Vrije U., Amsterdam, Vrije U., Amsterdam (2010).
- [2] J. A. N. van Tilburg, *Track Simulation and Reconstruction in LHCb* (Amsterdam : NIKHEF, 2005. - 166 p., 2005).
- [3] ATLAS Collaboration, *Search for the Standard Model Higgs boson in the $H \rightarrow WW^{(*)} \rightarrow l\nu l\nu$ decay mode with the ATLAS Detector*, ATLAS-CONF-2011-111 (2011).
- [4] S. Weinberg, *Cosmological Production of Baryons*, Phys. Rev. Lett. **42**, 10.1103/PhysRevLett.42.850 (1979).
- [5] K. Gizdov, *Studies of b Hadron Decays to Charmonium and the LHCb Upgrade and Operation*, Doktorska disertacija, The University of Edinburgh (2021), presented 17 Nov 2021.
- [6] M. Kobayashi in T. Maskawa, *CP Violation in the Renormalizable Theory of Weak Interaction*, Prog. Theor. Phys. **49**, 10.1143/PTP.49.652 (1973).
- [7] A. Mangoni, *Hadronic decays of the J/ψ meson*, Doktorska disertacija, Università' Di Perugia (2020), arXiv:2002.09675 [hep-ph] .
- [8] LHCb, *Future physics potential of LHCb*, LHCb-PUB-2022-012, CERN-LHCb-PUB-2022-012 (2022).
- [9] L. M. Garcia Martin (LHCb Collaboration), *Rare b -hadron decays at LHCb*, PoS **EPS-HEP2019**, 10.22323/1.364.0247 (2020).
- [10] CERN, *CERN homepage: <https://home.cern/>*, [ogled 8. 8. 2022].
- [11] L. Evans in P. Bryant, *LHC Machine*, Journal of Instrumentation **3** (08), S08001.
- [12] A. K. Morley, *Electron Bremsstrahlung Studies and Track Based Alignment of the ATLAS Detector*, Magistrsko delo, Melbourne U. (2010).
- [13] CERN, *LHCb homepage: <https://lhcb-outreach.web.cern.ch/>*, [ogled 8. 8. 2022].
- [14] M. Adinolfi *et al.* (LHCb RICH Group), *Performance of the LHCb RICH detector at the LHC*, Eur. Phys. J. C **73**, 2431 (2013).
- [15] Y. Li (LHCb UT Upgrade II team), *MAPS for the Upstream Tracker in LHCb Upgrade II*, Nucl. Instrum. Meth. A **1032**, 10.1016/j.nima.2022.166629 (2022).

- [16] LHCb, *LHCb Tracker Upgrade Technical Design Report*, CERN Document Server (2014).
- [17] P. A. Günther, *LHCb's Forward Tracking algorithm for the Run 3 CPU-based online track-reconstruction sequence*, LHCb-PROC-2022-009 (2022).
- [18] *Selected HLT2 reconstruction performance for the LHCb upgrade*: <https://lhcbpr-hlt.web.cern.ch/>, [ogled 15. 8. 2022].
- [19] *HLT2 Flame Graphs*: <https://lhcbpr-hlt.web.cern.ch/>, [ogled 15. 8. 2022].
- [20] T. Szumlak, *Events reconstruction at 30 MHz for the LHCb upgrade*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **936**, 356 (2019), frontier Detectors for Frontier Physics: 14th Pisa Meeting on Advanced Detectors.
- [21] E. Bos in E. Rodrigues, *The LHCb track extrapolator tools*, CERN-LHCB-2007-140, GLAS-PPE-2007-24 (2007).
- [22] P. Billoir, M. De Cian, P. Günther in S. Stemmler, *A parametrized Kalman filter for fast track fitting at LHCb*, Computer Physics Communications **265**, 108026 (2021).
- [23] LHCb, *LHCb magnet: Technical design report*, CERN-LHCC-2000-007 (2000).
- [24] *GitLab repository with Magnetic Field Map visualizer*: https://gitlab.cern.ch/kspenko/fieldmap_visualizer, [ogled 15. 8. 2022].
- [25] P. S. Maybeck, *Stochastic models, estimation and control. Volume 1*, Mathematics in science and engineering; v. 141 (Academic Press, 1979).
- [26] R. Barate *et al.* (LEP Working Group for Higgs boson searches, ALEPH, DELPHI, L3, OPAL), *Search for the standard model Higgs boson at LEP*, Phys. Lett. B **565**, 10.1016/S0370-2693(03)00614-2 (2003).
- [27] R. Frühwirth, *Application of Kalman filtering to track and vertex fitting*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **262**, 444 (1987).
- [28] R. Frühwirth in R. K. Bock, *Data analysis techniques for high-energy physics experiments*, ur. H. Grote, D. Notz in M. Regler, Zv. 11 (Cambridge University Press, 2000).
- [29] *Runge-Kutta method*, *Wikipedia*, [ogled 15. 8. 2022].
- [30] *Pierre_extrapolator*, *gitlab*: https://gitlab.cern.ch/ausachov/pierre_extrapolator/, [ogled 15. 8. 2022].
- [31] T. Sjostrand, S. Mrenna in P. Z. Skands, *PYTHIA 6.4 Physics and Manual*, JHEP **05** (FERMILAB-PUB-06-052-CD-T, LU-TP-06-13).
- [32] B. P. Kersevan in E. Richter-Was, *The Monte Carlo event generator AcerMC versions 2.0 to 3.8 with interfaces to PYTHIA 6.4, HERWIG 6.5 and ARIADNE 4.1*, Comput. Phys. Commun. **184**, 919 (2013).

-
- [33] M. Clemencic, G. Corti, S. Easo, C. R. Jones, S. Miglioranzi, M. Pappagallo in P. Robbe (LHCb), *The LHCb simulation application, Gauss: Design, evolution and experience*, J. Phys. Conf. Ser. **331**, 10.1088/1742-6596/331/3/032023 (2011).
- [34] D. J. Lange, *The EvtGen particle decay simulation package*, Nucl. Instrum. Meth. A **462**, 152 (2001).
- [35] S. A. et al., *Geant4—a simulation toolkit*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **506**, 250 (2003).
- [36] K. Kumericki, *Feynman Diagrams for Beginners*, ResearchGate (2016).
- [37] R. de la Madrid, *The decay widths, the decay constants, and the branching fractions of a resonant state*, Nucl. Phys. A **940**, 297 (2015).
- [38] prof. Pierre Billoir, *Internal communication and personal notes on the Parametrized Extrapolator* (2021), osebna komunikacija s prof. Pierrom Billoir in dostop do njegovih osebnih zapiskov.
- [39] K. Špenko, *Summer Student Presentation*, [ogled 8. 8. 2022] https://lhcb.web.cern.ch/Collaboration_prizes/.
- [40] R. Aaij, J. Albrecht, M. Belous, P. Billoir, T. Boettcher, A. Brea Rodríguez, D. Bruch, D. Pérez, A. Casais Vidal, D. Craik, P. Fernandez Declara, L. Funke, V. Gligorov, B. Jashal, N. Kazeev, D. Santos, F. Pisani, D. Pliushchenko, S. Popov in M. Williams, *Allen: A High-Level Trigger on GPUs for LHCb*, Computing and Software for Big Science **4** (2020).
- [41] M. Mihovilovič, *Sledenje nestabilnih delcev v magnetnih spektrometrih: diplomsko delo* ([M. Mihovilovič], 2007) bibliografija: str. 131-132.
- [42] M. Mihovilovic *et al.*, *Methods for Optical Calibration of the BigBite Hadron Spectrometer*, Nucl. Instrum. Meth. A **686**, 20 (2012).
- [43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot in E. Duchesnay, *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research **12**, 2825 (2011).
- [44] C. E. Rasmussen in C. K. I. Williams, *Gaussian processes for machine learning.*, Adaptive computation and machine learning (MIT Press, 2006) str. I–XVIII, 1–248.
- [45] K. P. Murphy, *Machine learning : a probabilistic perspective* (MIT Press, Cambridge, Mass. [u.a.], 2013).
- [46] J. Ver Hoef in N. Cressie, *Multivariable Spatial Prediction*, Mathematical Geology **25**, 219 (1993).
- [47] E. V. Bonilla, K. Chai in C. Williams, *Multi-task Gaussian Process Prediction*, v *Advances in Neural Information Processing Systems*, Zv. 20, ur. J. Platt, D. Koller, Y. Singer in S. Roweis (Curran Associates, Inc., 2007).

Literatura

- [48] C. R. H. et al., *Array programming with NumPy*, Nature **585**, 357 (2020).
- [49] C. A. et al., *Review of Particle Physics*, Physics Letters B **667**, 1 (2008), review of Particle Physics.

Dodatek A

Gibanje nabitih delcev v magnetnem polju

Trajektorija delca mora v statičnem magnetnem polju $\vec{B}(\vec{x})$ zadovoljiti enačbam gibanja, ki jih diktira **Lorentzova sila**. V odsotnosti električnega polja in ob predpostavki, da zanemarimo zavorno sevanje in vpliv snovi, dobimo iz Maxwellovih enačb, da je Lorentzova sila

$$\vec{F} = q\vec{v} \times \vec{B}, \quad (\text{A.1})$$

kjer je q naboj delca, v pa njegova hitrost z velikostjo $v = |\vec{v}|$ ($\vec{v} \equiv d\vec{x}/dt$). Iz nje dobimo za enačbe gibanja za delec v vakumu

$$\frac{d\vec{p}}{dt} = \frac{d(m\gamma d\vec{x}/dt)}{dt} = c^2 \kappa q \vec{v}(t) \times \vec{B}(\vec{x}(t)), \quad (\text{A.2})$$

kjer je $\vec{B}(\vec{x})$ statično magnetno polje, m je invariantna masa delca in \vec{x} je položaj delca v prostoru, c je hitrost svetlobe, t pa je čas v koordinatnem sistemu laboratorija. κ je proporcionalni faktor, ki je odvisen od izbire enot posameznih količin, kot bomo videli v nadaljevanju. Relativističen Lorentzov factor γ je podan kot

$$\gamma = (1 - \beta^2)^{-\frac{1}{2}}, \quad (\text{A.3})$$

kjer je

$$\beta = |\vec{\beta}| = \frac{v}{c}, \quad \vec{\beta} = \frac{\vec{v}}{c} = \frac{d\vec{x}}{d(ct)}, \quad (\text{A.4})$$

Enačbe gibanja A.2 lahko prepisemo v obliko, kjer časovna koordinata eksplicitno ne nastopa kot

$$\frac{d^2\vec{x}}{ds^2} = \left(\frac{\kappa q}{p}\right) \left(\frac{d\vec{x}}{ds}\right) \times \vec{B}(\vec{x}(s)), \quad (\text{A.5})$$

kjer je $s(t)$ koordinata vzdolž trajektorije (dolžina poti), z izrazom za hitrostjo $v = ds/dt$ in gibalno količino delca $\vec{p} = m\gamma\vec{\beta}c$ v laboratorijskem koordinatnem sistemu z $p = |\vec{p}| = m\gamma\beta c$.

Dokaz.

$$\frac{d(m\gamma d\vec{x}/dt)}{dt} \cdot \left(\frac{d\vec{x}}{dt}\right) \approx \vec{v} \cdot (\vec{v}(t) \times \vec{B}(\vec{x}(t))) \equiv 0, \quad (\text{A.6})$$

sledi iz enačbe A.2 in zato potem velja:

$$\left| \frac{d\vec{x}}{dt} \right| = v = \beta c = \text{konstanta} (\gamma \equiv \text{konstanta}), \quad (\text{A.7})$$

$$\frac{d\vec{x}}{dt} = \left(\frac{d\vec{x}}{ds} \right) \left(\frac{ds}{dt} \right) = \left(\frac{d\vec{x}}{ds} \right) \beta c, \quad (\text{A.8})$$

$$\frac{d^2\vec{x}}{dt^2} = \left(\frac{d^2\vec{x}}{ds^2} \right) \beta^2 c^2. \quad (\text{A.9})$$

□

V fiziki delcev se za standardne enote uporabljajo naslednje:

- q je večkratnik pozitivnega osnovnega naboja, torej je brezdimenzionalen,
- \vec{x} in s se izražata v metrih ampak tudi v centimetrih in milimetrih,
- p je v enotah GeV/c,
- \vec{B} je podan v T,
- κ je posledično sorazmeren s hitrostjo svetlobe v vakuumu in ima vrednost 0,299 792 458 (GeV/c)T⁻¹m⁻¹ (če so koordinate \vec{x} podane v npr. mm, se to spremeni tudi v κ).

V primeru eksperimentov s fiksnimi tarčami in posledično relativno majhnimi p_T/p_L (razmerje transverzalne komponente gibalne količine proti longitudinalni) vsaj v smeri streljanja delcev, je prikladno prepisati enačbe gibanja z odvodi vzdolž žarkovne osi z v obliko

$$\frac{d^2x}{dz^2} = \frac{q}{p} \frac{ds}{dz} \left[\frac{dx}{dz} \frac{dy}{dz} B_x - \left(1 + \left(\frac{dx}{dz} \right)^2 \right) B_y + \frac{dy}{dz} B_z \right], \quad (\text{A.10})$$

$$\frac{d^2y}{dz^2} = \frac{q}{p} \frac{ds}{dz} \left[-\frac{dx}{dz} \frac{dy}{dz} B_y + \left(1 + \left(\frac{dy}{dz} \right)^2 \right) B_x - \frac{dx}{dz} B_z \right]. \quad (\text{A.11})$$

kjer sedaj zapišemo

$$ds^2 = dx^2 + dy^2 + dz^2 \quad (\text{A.12})$$

$$\frac{ds}{dz} = \sqrt{1 + \left(\frac{dx}{dz} \right)^2 + \left(\frac{dy}{dz} \right)^2}. \quad (\text{A.13})$$

Integriranje enačb gibanj iz enačbe A.2 zahteva integracijo po šestih spremenljivkah (neznana gibalna količina p sledi iz ukrivljenosti). Z identiteto

$$\left(\frac{dx}{ds} \right)^2 + \left(\frac{dy}{ds} \right)^2 + \left(\frac{dz}{ds} \right)^2 \equiv 1 \quad (\text{A.14})$$

in pa poljubno izbiro ene koordinate (izbira referenčne ravnine pri specifični vrednosti z), dobimo samo pet prostih parametrov, ki popolnoma definirajo našo trajektorijo: dve koordinati, ki opišeta preboj referenčne ravnine pri $z_r = \text{konstanta}$, dve za smer trajektorije v tej točki in eno za gibalno količino[28].

Dodatek B

Izpeljava Kalmanovega filtra

Pri izpeljavi Kalmanovega filtra predpostavimo, da je posodobljen vektor stanja \vec{x}_k linearna kombinacija vektorja stanja napovedi $\vec{x}_{k|k-1}$ in residuala iz enačbe 3.11 podana kot

$$\begin{aligned}\vec{x}_k &= \vec{x}_{k|k-1} + K_k \vec{r}_{k|k-1} \\ &= \vec{x}_{k|k-1} + K_k(m_k - H_k \vec{x}_{k|k-1}),\end{aligned}\tag{B.1}$$

kjer je K_k matrika ojačenja. Cilj je poiskati optimalno vrednost K_k . Najprej definiramo napako med napovedanim in posodobljenim vektorjem stanja kot

$$\vec{e}_{k|k-1} = \vec{x}_{k|k-1} - \vec{x}_k^r,\tag{B.2}$$

$$\vec{e}_k = \vec{x}_k - \vec{x}_k^r,\tag{B.3}$$

kjer smo z \vec{x}_k^r označili neznano pravo vrednost vektorja stanja. Če sedaj vstavimo enačbo B.1 v enačbo B.3, dobimo za napako

$$\begin{aligned}\vec{e}_k &= \vec{x}_{k|k-1} + K_k(m_k - H_k \vec{x}_{k|k-1}) - \vec{x}_k^r \\ &= (\mathbb{I} - K_k H_k)(\vec{x}_{k|k-1} - \vec{x}_k^r) + K_k(m_k - H_k \vec{x}_k^r) \\ &= (\mathbb{I} - K_k H_k)\vec{e}_{k|k-1} + K_k \epsilon_k,\end{aligned}\tag{B.4}$$

kjer je ϵ_k merski šum, kateremu pripada kovariančna matrika $cov(\epsilon_k) = V_k$. Filtrirana posodobljena kovariančna matrika je potem enaka

$$C_k = cov(\vec{e})_k = (\mathbb{I} - K_k H_k)C_{k|k-1}(\mathbb{I} - K_k H_k)^\top + K_k V_k K_k^\top,\tag{B.5}$$

kjer je ta zveza znana pod imenom *Josephova formula*. Naslednji korak je minimizacija te kovariančne matrike. To lahko naredimo, tako da najprej izračunamo njeno sled

$$\begin{aligned}\text{Tr}(C_k) &= \text{Tr}(C_{k|k-1}) - \text{Tr}(K_k H_k C_{k|k-1}) - \text{Tr}(C_{k|k-1} (K_k H_k)^\top) \\ &\quad + \text{Tr}(K_k H_k C_{k|k-1} (K_k H_k)^\top) + \text{Tr}(K_k V_k K_k^\top) \\ &= \text{Tr}(C_{k|k-1}) - 2\text{Tr}(K_k H_k C_{k|k-1}) \\ &\quad + \text{Tr}(K_k (H_k C_{k|k-1} H_k^\top) K_k^\top) + \text{Tr}(K_k V_k K_k^\top),\end{aligned}\tag{B.6}$$

in jo nato odvajamo po K_k

$$\frac{\delta \text{Tr}(C_k)}{\delta K_k} = -2(H_k C_{k|k-1})^\top + 2K_k (H_k C_{k|k-1} H_k^\top) + 2K_k V_k,\tag{B.7}$$

kjer uporabimo naslednje zveze za sledi matrik

$$\frac{\delta \text{Tr}(AB)}{\delta A} = B^\top, \quad (\text{B.8})$$

$$\frac{\delta \text{Tr}(ABA^\top)}{\delta A} = 2AB \quad \text{za } B = B^\top. \quad (\text{B.9})$$

Na koncu, ker iščemo minimum, nastavimo vrednost odvoda B.7 na nič

$$-C_{k|k-1}H_k^\top + K_k(H_kC_{k|k-1}H_k^\top) + K_kV_k = 0, \quad (\text{B.10})$$

in s tem dobimo izraz za ojačitveno matriko kot

$$K_k = C_{k|k-1}H_k^\top (V_k + H_kC_{k|k-1}H_k^\top)^{-1}. \quad (\text{B.11})$$

Posodobljeno kovariančno matriko nato preko Josephove formule B.5 z uporabo enačbe B.10 dobimo z zvezo

$$\begin{aligned} C_k &= (\mathbb{I} - K_kH_k)C_{k|k-1}(\mathbb{I} - K_kH_k)^\top + (C_{k|k-1}H_k^\top - K_kH_kC_{k|k-1}H_k^\top)K_k^\top \\ &= (\mathbb{I} - K_kH_k)C_{k|k-1}(\mathbb{I} - K_kH_k)^\top + (\mathbb{I} - K_kH_k)C_{k|k-1}(K_kH_k)^\top \\ &= (\mathbb{I} - K_kH_k)C_{k|k-1}. \end{aligned} \quad (\text{B.12})$$

Več o izpeljavi Kalmanovega filtra je v knjigi[25].

Dodatek C

Izguba energije in sipanje

Med prehodom skozi material nabiti delci izgubljajo energijo. Ionizacija je za relativistične, nabite delce največji prispevek k tej izgubi. Dobro ga opišemo z *Bethe-Blockovo* formulo, ki se glasi:

$$-\left\langle \frac{dE}{dx} \right\rangle = 4\pi N_A r_e^2 m_e c^2 z^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 T_{max}}{(1-\beta^2)I^2} - \beta^2 - \frac{\delta}{2} \right], \quad (\text{C.1})$$

kjer je c hitrost svetlobe v vakuumu, r_e klasičen radij elektrona in m_e njegova masa, $\beta = \frac{v}{c}$, z pa naboj delca v enotah osnovnega naboja e . Z N_A smo označili Avogadrovo konstanto, Z vrstno število in A relativno atomsko maso. S T_{max} je označen maksimalen prenos energije pri trku, z I pa povprečni vzbuditveni potencial, δ pa je popravek zaradi gostote materiala. Formula je veljavna le za proste nabite delce, kar pomeni, da morajo imeti delci energijo večjo od ionizacijskega potenciala atoma $I = (10 \text{ eV})Z$. Količino $-dE/dx$ imenujemo moč ustavljanja. Tipično gledamo moč ustavljanja v odvisnosti od energije oz. gibalne količine delca. Kot lahko vidimo na sliki C.1 moč ustavljanja sprva pada kot β^{-2} proti minimumu preden začne naraščati logaritemsko. Za večino relativističnih delcev je moč ustavljanja približno enaka tej minimalni vrednosti, zato jih tipično imenujemo mip delci, torej delci, ki občutijo minimalno ionizacijo. Ta minimum močno zavisi od tipa materiala. Največji prispevek k izgubi energije za relativistične elektrone pa je v resnici **zavorno sevanje** (nem. *bremssstrahlung*). Seveda je zavorno sevanje prisotno pri vseh tipih delcev, vendar postane za mione in hadrone relevantno šele pri energijah v režimu TeV[49][2].

Pri rekonstrukciji trajektorij se privzame za večino delcev, da so mip. Zaradi tega lahko iz Bethe-Bloch enačbe C.1 zanemarimo odvisnost od β , zaradi česar se enačba močno poenostavi v

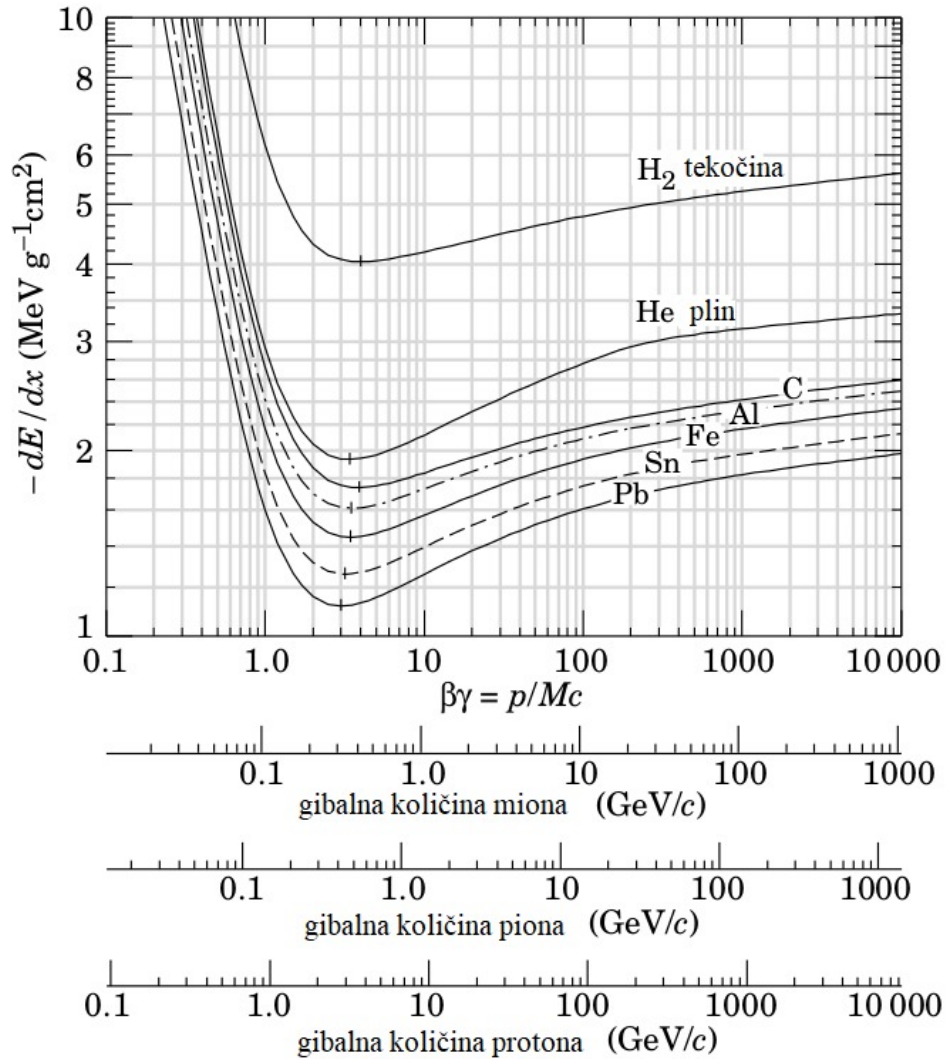
$$-\frac{dE}{dx} = c_{ion} \frac{Z}{A}, \quad (\text{C.2})$$

kjer smo z c_{ion} označili vse konstantne faktorje iz C.1. Za izgubo energijo vzdolž trajektorije dobimo prispevek

$$\Delta E = -c_{ion} \rho l \frac{Z}{A}, \quad (\text{C.3})$$

kjer je ρ gostota materiala in l dolžina, ki jo je delec prepotoval v materialu. V primeru elektronov je potrebno zaradi zavornega sevanja uporabiti drugačen popravek, ki nam da končen izraz

$$\Delta E = -E(1 - e^{-\frac{l}{X_0}}), \quad (\text{C.4})$$



Slika C.1: Graf moči ustavljanja $-dE/dx$ kot funkcija gibalne količine prikazana za različne materiale[2].

kjer je X_0 karakteristična sevalna dolžina materiala. Pri elektronih pa je potrebno zaradi izgube energije popraviti ne samo zadnjo komponento vektorja stanja ampak tudi njegovo kovariančno matriko. Dobimo dodaten prispevek v matriki modelskega šuma podan kot

$$Q_{55} = \left(\frac{q}{p}\right)^2 \left(e^{-\frac{\ln 3}{\ln 2} \frac{l}{X_0}} - e^{-2 \frac{l}{X_0}} \right). \quad (\text{C.5})$$

Ta popravek poveča napovedano napako gibalne količine oz. ukrivljenosti trajektorije elektronov pri prehodu čez vsako plast materiala[2].

Med prehodom delca skozi snov prihaja do različnih procesov. Med drugim prihaja tudi do sipanja delcev skozi snov. V primeru Coulombovega sipanja si lahko pomagamo z t.i. modelom MSC - **večkratnega Coulombovega sipanja** (ang. *multiple scattering*), s katerim obravnavamo sipanje prehajajočega delca na zveznem delu volumna. Formalizem Moliere-Bethe dobro opisuje večkratno sipanje tako pri majhnih kot tudi pri velikih kotih. Večino (98 %) odbitih trajektorij lahko dobro opišemo z Gaussovo porazdelitvijo, iz katere lahko nato ocenimo standardno

deviacijo (ang. *root-mean-squared*), za katero imamo podano fizikalno zvezo preko Highland-Lynch-Dahl formule[2]:

$$\theta_0 = \frac{13.6\text{MeV}}{\beta c p} \sqrt{\frac{\Delta z \sqrt{1 + t_x^2 + t_y^2}}{X_0}} \left[1 + 0.038 \ln \frac{\Delta z \sqrt{1 + t_x^2 + t_y^2}}{X_0} \right]. \quad (\text{C.6})$$

kjer je Δz debelina materiala z X_0 pa smo zopet označili karakteristično sevalno dolžino materiala. Zgornja zveza nam pove, da ima delec z energijo 13,6 GeV povprečen sipalni kot 1 mrad na večkratnik sevalne dolžine X_0 . [49][2][1].

Ker je večkratno sevanje naključen proces, ne moremo napovediti popravka k vrednosti vektorjev stanj. Zaradi tega večkratno sipanje prispeva zgolj k nedoločeni trajektorije z dodatnim šumom v matriki modelskega šuma Q . V primeru tankega sipalnega materiala dobimo popravke k naslednjim komponentam matrike

$$Q_{33} = (1 + t_x)^2 (1 + t_x^2 t_y^2) \theta_0, \quad (\text{C.7})$$

$$Q_{44} = (1 + t_y)^2 (1 + t_x^2 t_y^2) \theta_0, \quad (\text{C.8})$$

$$Q_{34} = t_x t_y (1 + t_x^2 t_y^2) \theta_0, \quad (\text{C.9})$$

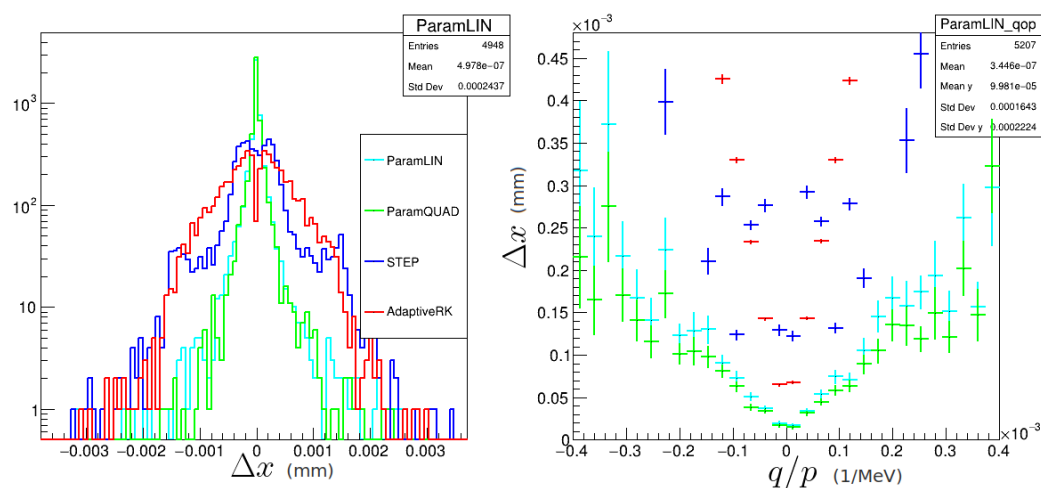
kjer je θ_0 sipalni kot, ki ga dobimo iz Moliere-Bethe formalizma. V praksi pa predpostavka tankega sipalnega sloja ne velja in je potrebno upoštevati večkratno sipanje kot zvezen proces. Prispevek k matriki Q je simetrična matrika in se za zvezen prehod glasi

$$Q(\Delta z) = \begin{pmatrix} Q_{33} \frac{\Delta z^2}{3} & Q_{34} \frac{\Delta z^2}{3} & Q_{33} \frac{\Delta z}{2} & Q_{34} \frac{\Delta z}{2} & 0 \\ \cdots & Q_{44} \frac{\Delta z^2}{3} & Q_{34} \frac{\Delta z}{2} & Q_{44} \frac{\Delta z}{2} & 0 \\ \cdots & \cdots & Q_{33} & Q_{34} & 0 \\ \cdots & \cdots & \cdots & Q_{44} & 0 \\ \cdots & \cdots & \cdots & \cdots & 0 \end{pmatrix}. \quad (\text{C.10})$$

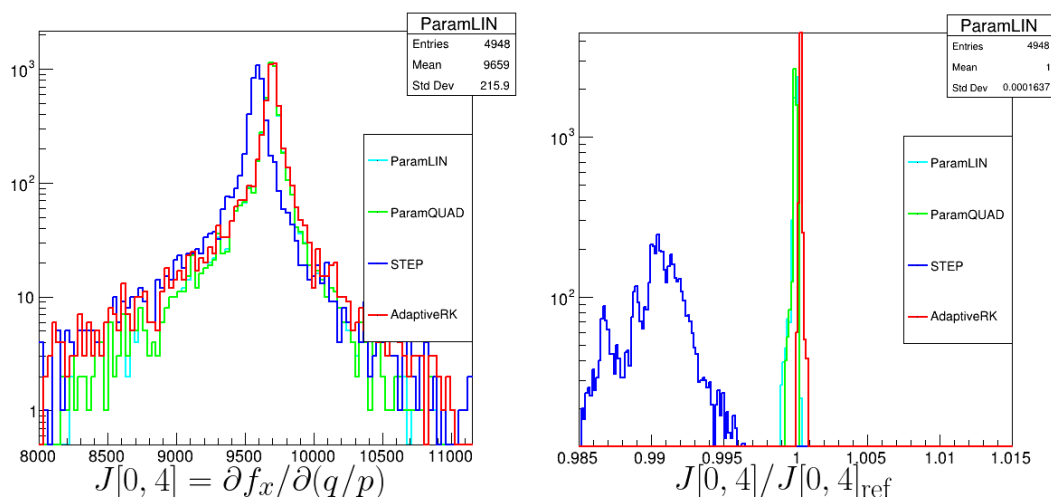
Popravek za sipanje skozi debelo sipalno plast moramo upoštevati, kadar je ta debelina istega reda velikosti kot dolžina posamezne ekstrapolacije. Najbolj očiten tak primer je sipanje skozi zrak pri ekstrapolacijah čez magnet. V LHCb rekonstrukcijskem programu v praksi predpostavimo zgolj debele sipalne plasti, saj aproksimacija tankega sipanja ponavadi podceni prispevek k nedoločeni.

Dodatek D

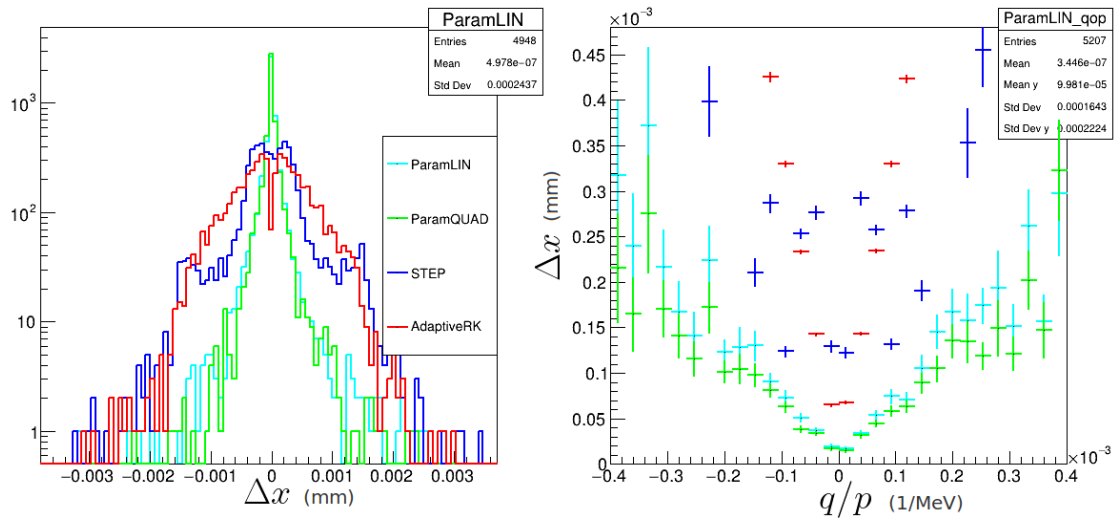
Dodatni testi parametriziranega ekstrapolatorja



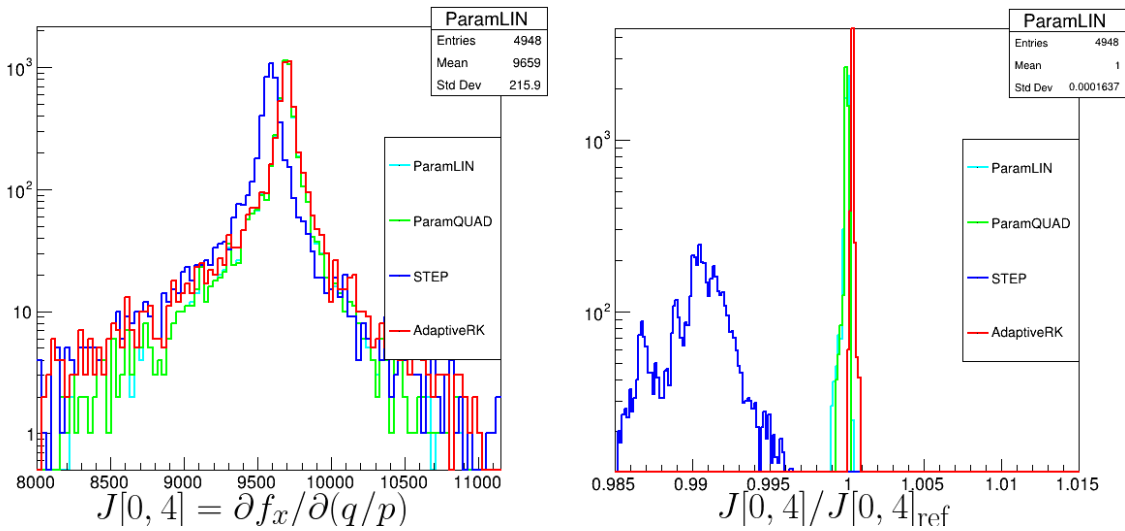
Slika D.1: Študija natančnosti parametriziranega ekstrapolatorja pri ekstrapolacijah v smeri od detektorja VELO do UT.



Slika D.2: Študija natančnosti Jacobijeve matrike parametriziranega ekstrapolatorja pri ekstrapolacijah v smeri od detektorja VELO do UT.



Slika D.3: Študija natančnosti parametriziranega ekstrapolatorja pri ekstrapolacijah v smeri od detektorja UT do VELO.



Slika D.4: Študija natančnosti Jacobijeve matrike parametriziranega ekstrapolatorja pri ekstrapolacijah v smeri od detektorja UT do VELO.