

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS  
TAIKOMOSIOS MATEMATIKOS KATEDRA

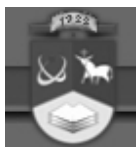
Vincas Benevičius

**JUDESIO TRAJEKTORIJOS ATKŪRIMAS BEI  
DEFORMACIJŲ FIKSAVIMAS REMIANTIS  
PAGREIČIAIS**

Magistro darbas

Darbo vadovas  
doc. dr. N. Listopadskis

KAUNAS, 2007



KAUNO TECHNOLOGIJOS UNIVERSITETAS  
 FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS  
 TAIKOMOSIOS MATEMATIKOS KATEDRA

TVIRTINU  
 Katedros vedėjas  
 prof. dr. J.Rimas  
 2007 06 06

**JUDESIO TRAJEKTORIJOS ATKŪRIMAS BEI  
 DEFORMACIJŲ FIKSAVIMAS REMIANTIS  
 PAGREIČIAIS**

Magistro darbas

Vadovas

( ) doc. dr. N. Listopadskis

2007 06 06

Recenzentas doc. dr. Dalius Rubliauskas

( )

2007 06 06

Atliko

FMMM-5 gr. stud.

( ) V. Benevičius

2007 06 06

KAUNAS, 2007

**KVALIFIKCINĖ KOMISIJA**

Pirmininkas: Leonas Saulis, profesorius (VGTU)

Sekretorius: Eimutis Valakevičius, docentas (KTU)

Nariai: Algimantas Jonas Aksomaitis, profesorius (KTU)

Vytautas Janilionis, docentas (KTU)

Vidmantas Povilas Pekarskas, profesorius (KTU)

Rimantas Rudzkis, habil.dr., banko „NORD/LB“ vyriausiasis analitikas (MII)

Zenonas Navickas, profesorius (KTU)

Arūnas Barauskas, UAB „Elsis“ generalinio direktoriaus pavaduotojas

**Benevičius, V. Judesio trajektorijos atkūrimas bei deformacijų fiksavimas remiantis pagreičiais: taikomosios matematikos magistro darbas / darbo vadovas doc. dr. N. Listopadskis, taikomosios matematikos katedra, fundamentaliųjų mokslų fakultetas, Kauno technologijos universitetas. Kaunas, 2007. - 62 p.**

### **SANTRAUKA**

Šiame darbe nagrinėjama trajektorijos atkūrimo bei deformacijų fiksavimo remiantis pagreičiais problema. Uždavinys sprendžiamas analizuojant dviejų taškų judėjimo trajektorijas plokštumoje.

Analizuojami ir palyginami skaitinio integravimo metodai: integravimas naudojant kubinį interpoliacinį spliną, trečios eilės B splinų tiesinius darinius, Niutono ir Koteso formules. Pagrindžiamas Niutono ir Koteso formulų integravimui pasirinkimas. Duomenys analizei yra modeliuojami, kadangi realių dviejų taškų sistemos pagreičių verčių, išmatuotų pasirinktu akselerometru, be finansinių investicijų gauti neįmanoma.

Darbe tiriamos paklaidų priklausomybės nuo pasirinktų integravimo metodo parametrų, pasirinkto akselerometro darbinių parametrų, nagrinėjama paklaidų augimo dinamika pasirinktoms judesio trajektorijoms, sudaromas kriterijus deformacijų fiksavimui.

Šis darbas yra pristatytas trijose konferencijose (VI studentų konferencija, 2006; Matematika ir matematinis modeliavimas, 2006; Lietuvos matematikų draugijos XLVII konferencija, 2006). Taip pat išleisti trys straipsniai šios temos pagrindu [6], [7], [8] ir pateiktas vienas naujas Lietuvos matematikų draugijos XLVIII konferencijai.

**Benevičius V. Pathway restoration and deformation sensing using acceleration data : Master's work in applied mathematics / supervisor dr. assoc. prof. N. Listopadskis; Department of Applied mathematics, Faculty of Fundamental Sciences, Kaunas University of Technology. – Kaunas, 2006. – 62 p.**

### **SUMMARY**

This work is targeted towards pathway restoration and deformation sensing using acceleration data. This problem is solved analyzing two point system movements in a plane.

Integration methods are compared, choice of Newton - Kotess formulas for integration is motivated. Data for analysis is provided by simulation, because real data acquisition requires financial investments.

Error dependencies on integration parameters, accelerometer operation parameters are analyzed. The dynamics of error development while observing different pathways is being tracked; a criterion for sensing deformation is defined.

This work has been reported at the three conferences (6th Student's Conference, 2006; Mathematics and Mathematical Modeling, 2006; 47th Conference of Lithuanian Mathematician Association, 2006). In additions, three articles have been published based on this topic [6], [7], [8] and one more article is going to be presented at the 48th Conference of Lithuanian Mathematician Association.

## TURINYS

<b>IVADAS</b> .....	<b>10</b>
<b>1 BENDROJI DALIS</b> .....	<b>11</b>
1.1 AKSELEROMETRAI .....	11
1.1.1 Akselerometro veikimo principai .....	11
1.1.2 Akselerometrų tipai ir taikymo sritys .....	14
1.2 SKAITINIO INTEGRAVIMO METODAI .....	18
1.2.1 Niutono ir Koteso kvadratūrinės formulės .....	18
1.2.2 Funkcijų, pateiktų reikšmių lentelė, integravimas .....	19
1.3 LAISVAS ROTACIJOS CENTRAS .....	20
<b>2 TIRIAMOJI DALIS</b> .....	<b>21</b>
2.1 DARBO TIKSLAS .....	21
2.2 MATEMATINIS MODELIS, REIKALAVIMAI IR PRIELAIDOS .....	21
2.3 DUOMENŲ GENERAVIMAS IR PARUOŠIMAS .....	22
2.4 INTEGRAVIMO METODO PASIRINKIMAS .....	25
2.5 INTEGRAVIMO STRATEGIJA .....	27
2.6 TRAJEKTORIJOS ATKŪRIMAS. PAKLAIDOS .....	32
2.7 DEFORMACIJŲ FIKSAVIMAS .....	39
<b>3 PROGRAMINĖ REALIZACIJA IR INSTRUKCIJA VARTOTOJUI</b> .....	<b>48</b>
<b>4 DISKUSIJA</b> .....	<b>50</b>
<b>IŠVADOS</b> .....	<b>52</b>
<b>REKOMENDACIJOS</b> .....	<b>53</b>
<b>LITERATŪRA</b> .....	<b>54</b>
<b>1 PRIEDAS. PROGRAMOS TEKSTAS</b> .....	<b>55</b>

**LENTELIŲ SĄRAŠAS**

2.2.1 lentelė .....	23
MXD2020G&M akselerometrų tikslumas esant skirtingiems darbo parametrams .....	23
2.3.1 lentelė .....	27
Metodų paklaidos .....	27
2.4.1 lentelė .....	28
Paklaidų tyrimo rezultatai .....	28
2.4.2 lentelė .....	30
Paklaidų priklausomybė nuo akslerometro parametrų .....	30
2.4.3 lentelė .....	30
Paklaidų priklausomybė nuo interguojamo intervaliuko dydžio.....	31
2.5.1 lentelė .....	38
Paklaidos praėjus vienai valandai .....	38
2.6.1 lentelė .....	48
Deformacijos fiksavimo laikai, kai deformacijos pradžia yra per nagrinėjamo intervalo vidurį.....	48

## PAVEIKSLŲ SĄRAŠAS

1.1.1.1 pav. Spyruoklės – svarelis sistema .....	11
1.1.1.2 pav. Svarelis svyravimų slopimas laike .....	12
1.1.1.3 pav. Sistema esant stalo vibracijai .....	12
1.1.1.4 pav. Tikrojo ir spėjamo nuokrypio amplitudžių grafikai .....	13
1.1.2.1 pav. 3-jų kristalų pjezoelektrinis akselerometras. Tokia konfigūracija minimizuoja trumpalaikius šiluminius efektus ir bazės apkrovą .....	15
1.1.2.2 pav. Naudinga dažnių juosta tipiniam pjezoelektriniam akselerometrui logaritminiam grafike. ....	15
1.1.2.3 pav. Pjezo pasipriešinimo akselerometro struktūra .....	16
1.1.2.4 pav. Talpinis akselerometras. Esant pagreičiui dėl masės atstumo pasikeitimo iki plokštelių, keičiasi talpa .....	17
2.1.1 pav. Dviejų taškų (akselerometrų) sistema .....	21
2.2.1 pav. Darbe nagrinėjamos trajektorijos. $Ax_1(t)$ ir $Ay_1(t)$ atitinka 2.2.1 formulę, $Ax_2(t)$ ir $Ay_2(t)$ - 2.2.2, $Ax_3(t)$ ir $Ay_3(t)$ - 2.2.3, $Bx_1(t)$ ir $By_1(t)$ - 2.2.4, $Bx_2(t)$ ir $By_2(t)$ - 2.2.5, $Bx_3(t)$ ir $By_3(t)$ - 2.2.6. ....	24
2.2.2 pav. Duomenų generavimo schema .....	25
2.3.1 pav. Funkcijos $y(x) = \sin(x)$ integravimo paklaidos. 1 – naudojant kubinį interpoliacinį splainą, 2 – naudojant tiesinį kubinių B splainų darinį, 3 – remiantis Niutono ir Koteso formulėmis. ....	26
2.3.2 pav. Funkcijos $y(x) = \sin(x) + \sin(2x)$ integravimo paklaidos. 1 – naudojant kubinį interpoliacinį splainą, 2 – naudojant tiesinį kubinių B splainų darinį, 3 – remiantis Niutono ir Koteso formulėmis. ....	26
2.3.3 pav. Funkcijos $y(x) = \sqrt{x}$ integravimo paklaidos. 1 – naudojant kubinį interpoliacinį splainą, 2 – naudojant tiesinį kubinių B splainų darinį, 3 – remiantis Niutono ir Koteso formulėmis. ....	27
2.4.1 pav. Integravimo strategija .....	28
2.4.2 pav. Paklaidų priklausomybė nuo n trajektorijai $A_1+B_1$ .....	29
2.4.3 pav. Paklaidų priklausomybė nuo n trajektorijai $A_2+B_2$ .....	30
2.5.1 pav. Atkurta trajektorija $A_1+B_1$ .....	32
2.5.2 pav. Paklaidos taškui B trajektorijoje $A_1+B_1$ .....	33
2.5.3 pav. Atkurta trajektorija $A_1+B_2$ .....	33
2.5.4 pav. Paklaidos taškui B trajektorijoje $A_1+B_2$ .....	33
2.5.5 pav. Atkurta trajektorija $A_1+B_3$ .....	34
2.5.6 pav. Paklaidos taškui B trajektorijoje $A_1+B_3$ .....	34
2.5.7 pav. Atkurta trajektorija $A_2+B_1$ .....	34
2.5.8 pav. Paklaidos taškui B trajektorijoje $A_2+B_1$ .....	35
2.5.9 pav. Atkurta trajektorija $A_2+B_2$ .....	35
2.5.10 pav. Paklaidos taškui B trajektorijoje $A_2+B_2$ .....	35
2.5.11 pav. Atkurta trajektorija $A_2+B_3$ .....	36
2.5.12 pav. Paklaidos taškui B trajektorijoje $A_2+B_2$ .....	36
2.5.13 pav. Atkurta trajektorija $A_3+B_1$ .....	36
2.5.14 pav. Paklaidos taškui B trajektorijoje $A_3+B_1$ .....	37
2.5.15 pav. Atkurta trajektorija $A_3+B_2$ .....	37
2.5.16 pav. Paklaidos taškui B trajektorijoje $A_3+B_2$ .....	37
2.5.17 pav. Atkurta trajektorija $A_3+B_3$ .....	38
2.5.18 pav. Paklaidos taškui B trajektorijoje $A_3+B_3$ .....	38
2.5.19 pav. Pagreičių funkcijų grafikai trajektorijoms $A_1+B_1$ ir $A_2+B_2$ .....	39
2.6.1 pav. Paklaidų grafikas trajektorijai $A_1+B_1$ iki $t = 5$ s .....	40
2.6.2 pav. Paklaidų grafikas trajektorijai $A_1+B_1$ $t = [2,5;4]$ .....	40
2.6.3 pav. Paklaidų grafikas trajektorijai $A_1+B_1$ iki $t = 5$ s deformacijai įvykus laiko momentu $t = 3$ s ...	41



2.6.4 pav. Paklaidų grafikas trajektorijai A1+B1 $t = [2,5;4]$ deformacijai įvykus laiko momentu $t = 3$ s..	41
2.6.5 pav. Paklaidų grafikas trajektorijai A1+B1 $t = [3;3,5]$ .....	42
2.6.6 pav. Paklaidų grafikas trajektorijai A1+B1 $t = [3;3,5]$ deformacijai įvykus laiko momentu $t = 3$ s...	42
2.6.7 pav. $f(t_k)$ grafikas trajektorijai A1+B1 iki $t = 5$ s.....	43
2.6.8 pav. $f(t_k)$ grafikas trajektorijai A1+B1 iki $t = 5$ s deformacijai įvykus laiko momentu $t = 3$ s.....	44
2.6.9 pav. $f(t_k)$ grafikas trajektorijai A1+B1 iki $t = 100$ s.....	45
2.6.10 pav. $f(t_k)$ grafikas trajektorijai A1+B1 iki $t = 100$ s deformacijai įvykus laiko momentu $t = 80$ s. .	45
2.6.11 pav. $s(t_i)$ grafikas trajektorijai A1+B1 iki $t = 5$ s be deformacijos ir deformacijai įvykus laiko momentu $t = 3$ s.....	46
2.6.12 pav. $s(t_i)$ grafikas trajektorijai A2+B2 iki $t = 5$ s be deformacijos ir deformacijai įvykus laiko momentu $t = 3$ s.....	46
2.6.13 pav. $s(t_i)$ grafikas trajektorijai A3+B3 iki $t = 5$ s be deformacijos ir deformacijai įvykus laiko momentu $t = 3$ s.....	47
2.6.14 pav. $s(t_i)$ grafikas trajektorijai A1+B1 iki $t = 500$ s be deformacijos ir deformacijai įvykus laiko momentu $t = 400$ s.....	47
2.6.15 pav. $s(t_i)$ grafikas trajektorijai A2+B2 iki $t = 500$ s be deformacijos ir deformacijai įvykus laiko momentu $t = 400$ s.....	47
2.6.16 pav. $s(t_i)$ grafikas trajektorijai A2+B2 iki $t = 500$ s be deformacijos ir deformacijai įvykus laiko momentu $t = 400$ s.....	48
3.6.1 pav. Programos pagrindinis darbo langas .....	49

## Įvadas

Šiame darbe nagrinėsime dviejų taškų sistemos judesį, kai gaunami duomenys yra taškų pagreičių vektoriai pastoviu laiko žingsniu. Tai vienas pirmųjų bandymų tirti judesį remiantis vien tik pagreičiais, kurie pastoviu laiko žingsniu yra pamatuojami akcelerometru. Remiantis tokiais matavimais bandysime atkurti judesio trajektoriją bei užfiksuoti deformacijos atsiradimo sistemoje pradžios momentą.

Trajektorijos atkūrimo galimybė leistų tokias nedideles ir pigias akcelerometrų sistemas naudoti vietoje nors ir tikslesnių, tačiau ir gerokai brangesnių GPS sistemų, skirtų nustatyti objekto padėtį. Deformacijos fiksavimas savo ruožtu gali būti sėkmingai naudojamas didelių judančių ar stacionarių, tačiau su gana nemaža dinamika, objektų, tokių kaip didieji laivai ar aukšti pastatai, geometrijos stebėjimui tokiu būdu kontroliuojant apkrovas (laivo atveju) ar išvengiant nelaimių (pastatų atveju) dėl, tarkim, stiprėjančio vėjo ar žemės drebėjimo. Apskritai tokių sistemų pritaikymo galimybės yra labai plačios: pradedant buitimi (pvz. automobiliuose) ir baigiant karine pramone (pvz. raketos valdymo blokuose).

Darbe aptarsime aparatūrinės įrangos įvairovę, jos veikimo principus, matavimus įtakojančius faktorius. Taip pat apibrėšime matematinę problemos modelį, bandysime pagrįstai pasirinkti skaitinio integravimo metodą, apibrėšime duomenų generavimo algoritmą ir tirsime paklaidų priklausomybę nuo įvairių faktorių, tokių kaip aparatūrinės įrangos (akcelerometrų) darbo, skaičiavimo metodo parametrų. Bandysime sudaryti kriterijų, kuriuo remiantis būtų galima fiksuoti deformacijos pradžios momentą.

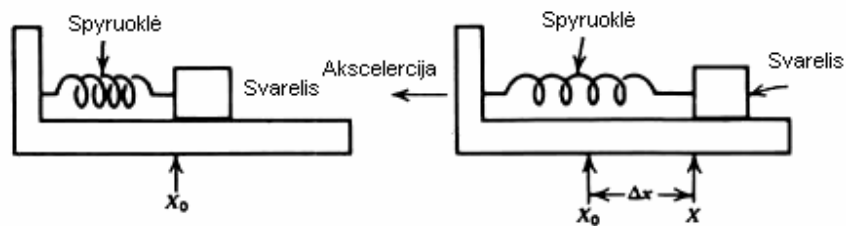
## 1 Bendroji dalis

### 1.1 Akselerometrai

#### 1.1.1 Akselerometro veikimo principai

##### Spyruoklės – svarelio sistema

Niutono dėsnis teigia, kad masės  $m$  kūnas turi pagreitį  $a$ , tuomet kūną veikia jėga  $F = ma$ . Huko dėsnis teigia, kad jei spyruoklės standumo koeficientas yra  $k$ , tuomet spyruoklę, pajudintą iš pusiausvyros padėties atstumu  $dx$ , tuomet tą spyruoklę veikia jėga  $F = k \cdot dx$ .



1.1.1.1 pav. Spyruoklės – svarelio sistema

1.1.1.1 paveiksle turime svarelį, kuris gali laisvai slysti paviršiumi. Svarelį su baze jungia spyruoklė. Įvykus akceleracijai, svarelis pasislenka postūmiu  $\Delta x$  ir atsiradusioms jėgoms pagal minėtus dėsnius galioja lygybė:

$$ma = k\Delta x \quad (1.1.1.1)$$

1.1.1.1 formulė leidžia pagreičio matavimą pakeisti spyruoklės ilgio pokyčio matavimu ir taip gauti pagreitį:

$$a = \frac{k}{m} \Delta x \quad (1.1.1.2)$$

Pastaroji formulė yra ryšys tarp spyruoklės ilgio pokyčio ir pagreičio.

##### Savasis dažnis ir svyravimų amplitudės slopimas

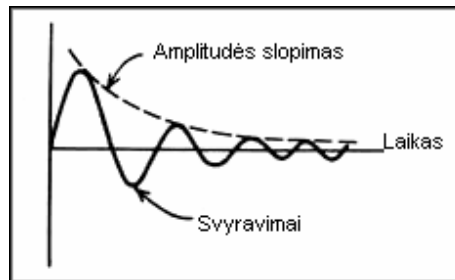
Atidžiau patyrinęję tik apibrėžtą principą, mes pastebime dar vieną spyruoklės – svarelio sistemos charakteristiką, kuri komplikuoja analizę. Kiekviena tokia sistema svyruoja tam tikru *savuoju dažniu*. Patirtis rodo, kad jei patrauksime svarelį ir jį paleisime, pastarasis pradės svyruoti ir vien tik trinties jėga ilgai privers svarelį sustoti. Bet koks ilgio pokyčio matavimas reaguos į minėtą svyravimą tarsi vykėtų sistemos akceleracija. Savasis dažnis:

$$f_N = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad (1.1.1.3)$$

Trintis, ilginiui priverčianti svarelį sustoti apibrėžiama slopinimo koeficientu. Bendru atveju, toks svyravimo efektas dar vadinamas *laikinu atsaku*, nusakomu periodiniu silpstančiu signalu, kuris aprašomas šia lygtimi:

$$x_T(t) = x_0 e^{-\mu t} \sin(2\pi f_N t) \quad (1.1.1.4)$$

Čia  $\mu$  - slopinimo koeficientas,  $x_0$  - maksimalus (pradinis) nuokrypis.



**1.1.1.2 pav. Svarelįo svyravimų slopinimas laike**

Pastarojoje išraiškoje naudojami dydžiai  $\mu$  (slopinimo koeficientas) ir  $f_n$  (savasis dažnis) turi didelį poveikį akselerometrų taikyme.

### Vibracijos

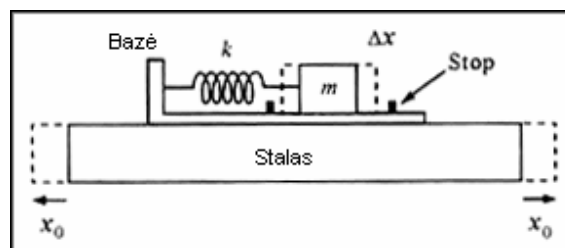
Savojo dažnio ir amplitudės slopinimo poveikis svarelįo – spyruoklės sistemos elgesiui geriausiai parodomas pritaikius priverstinę vibraciją. Jei sistema priversta vibruoti, tuomet gauta bazės akceleracija aprašoma lygtimi:

$$a(t) = -\omega^2 x_0 \sin \omega t \quad (1.1.1.5)$$

Jei ši išraiška panaudojama 1.1.1 formulėje, tuomet svarelįo judesys aprašomas:

$$\Delta x = -\frac{m x_0}{k} \omega^2 \sin \omega t \quad (1.1.1.6)$$

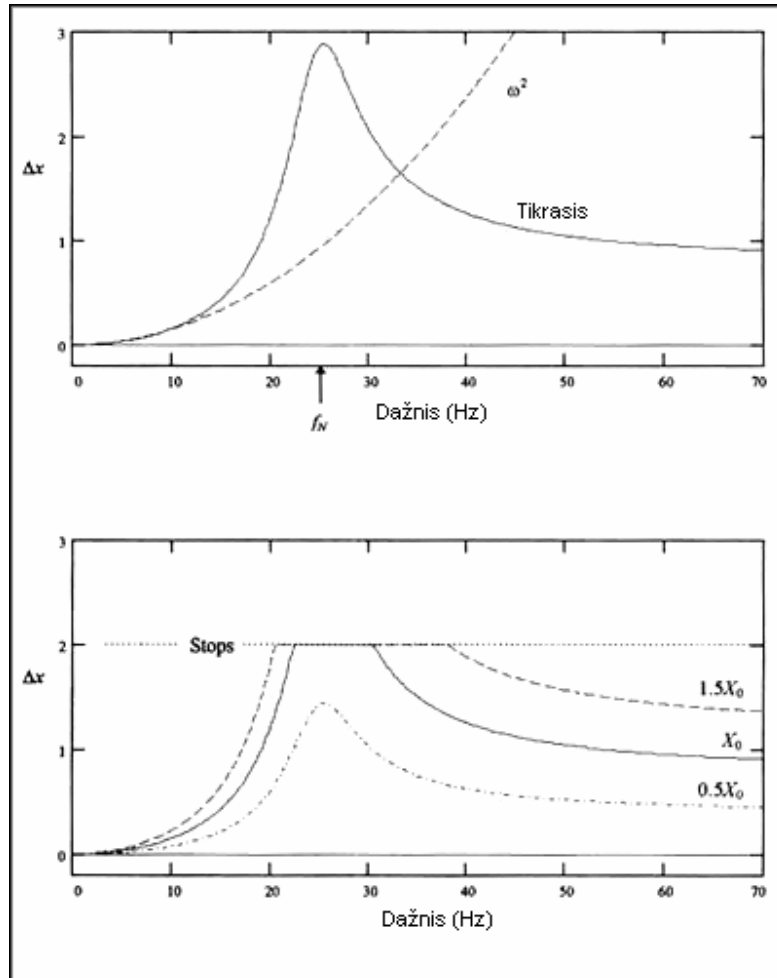
Čia  $\omega = 2\pi f$ , kur  $f$  - priverstinis dažnis (vibracija).



**1.1.1.3 pav. Sistema esant stalo vibracijai**

Tarkime turime situaciją, pavaizduotą pateiktame paveiksliuke. Spyruoklės – svarelįo sistema pritvirtinta prie vibruojančio stalo. 1.1.1.6 formulėje  $x_0$  yra stalo vibracijos pikas, o  $\Delta x$  - svarelįo

padėties pokytis. Remiantis 1.1.1.6 turime, kad svarelčio svyravimų amplitudė su vibracijos dažniu susijusi kvadratine priklausomybe, o su stalo svyravimu amplitudė – tiesiškai. Tačiau toks rezultatas gaunamas nevertinant pačios sistemos savojo svyravimų dažnio.



#### 1.1.1.4 pav. Tikrojo ir spėjamo nuokrypio amplitudžių grafikai

1.1.1.2 paveiksluke matomas tikrojo svyravimo amplitudės grafikas esant stalo vibracijai bei pagal 1.1.1.6 formulę įvertinto svyravimo amplitudės grafikas. Matome, kad kai savasis sistemos dažnis sutampa su stalo vibracijos dažniu, turime rezonansą. Rezonanso amplitudę sąlygoja slopimo stiprumas. Spyruoklės – svarelčio sistemos vibracija aprašoma 1.1.1.6 išraiška tik iki dažnio, apytiksliai lygaus  $\frac{f_N}{2.5}$ .

1.1.1.2 paveiksle taip pat matome, kad dažniams, gerokai didesniems nei savasis dažnis, masės judesys proporcingas stalo judesio pikui  $x_0$ , bet ne dažniui. Todėl sistema tampa poslinkio sensoriumi.

Bendru atveju, akselerometrai nenaudojami esant dažniams, artimiems jų saviesiems dažniams dėl didelio netiesiškumo gaunamiems duomenims.

### 1.1.2 Akselerometrų tipai ir taikymo sritys

Įvairiose srityse naudojami skirtingų tipų akselerometrai. Tipo pasirinkimas priklauso nuo matuojamo pagreičio diapazono, savojo dažnio ir slopimo koeficiento. Esminis skirtumas tarp akselerometrų tipų yra tik masės poslinkio matavimas. Bendru atveju iš specifikacijų lentelių galima sužinoti savąjį dažnį, slopimo koeficientą ir mastelį, kuris sieja įėjimo duomenis su išėjimo duomenimis (dažniausiai įtampa).

#### Potenciometrinis akselerometras

Paprasčiausio tipo akselerometras, matuoja masės poslinkį sujungiant masę su potenciometro alkūne. Tokiu būdu masės padėtis išreiškiama kaip kintantis pasipriešinimas. Savasis šio tipo akselerometrų dažnis dažniausiai neviršija 30Hz, taip apribodamas tokio tipo akselerometrų naudojimą iki pastovios būsenos akceleracijos matavimo ar nedidelio dažnio svyravimų matavimo.

#### LVDT akselerometrai

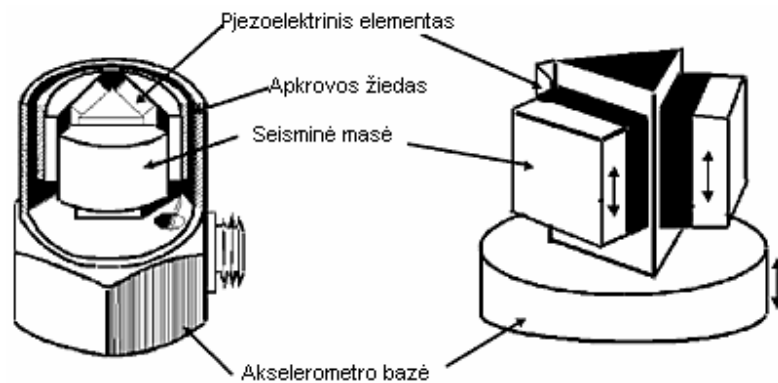
Kitas akselerometrų tipas naudoja LVDT (Linear Variable Differential Transformer), kuris ir atstoja seisminę masę. Ašies poslinkis iš karto transformuojamas į tiesiškai pagreičiui proporcingą įtampą. Šio tipo akselerometrų savasis dažnis dažniausiai neviršija 80Hz ir dažniausiai naudojami pastovios būsenos pagreičiams matuoti ar mažo dažnio svyravimams matuoti.

#### Kintamos magnetinės varžos akselerometrai

Šio tipo akselerometrai papuola į tą pačią kategoriją, kaip ir LVDT akselerometrai, nes abiem atvejais panaudojamas indukcijos reiškiny. Šiame akselerometre masę atstoja nuolatinis magnetas. Matavimas atliekamas remiantis indukuota riteje įtampa, kuomet magnetinė masė juda. Šio tipo akselerometrai naudojami tik vibracijai ar šokui matuoti, nes jie grąžina signalą tik tada, kai masė juda su pagreičiu. Savasis dažnis tipiškai yra mažesnis nei 100Hz. Šio tipo akselerometrai dažnai naudojami alyvos paieškose siekiant aptikti vibracijas, atspindėtas nuo požeminio uolienų sluoksnio. Šiame kontekste akselerometras dažnai vadinamas geo – telefonu.

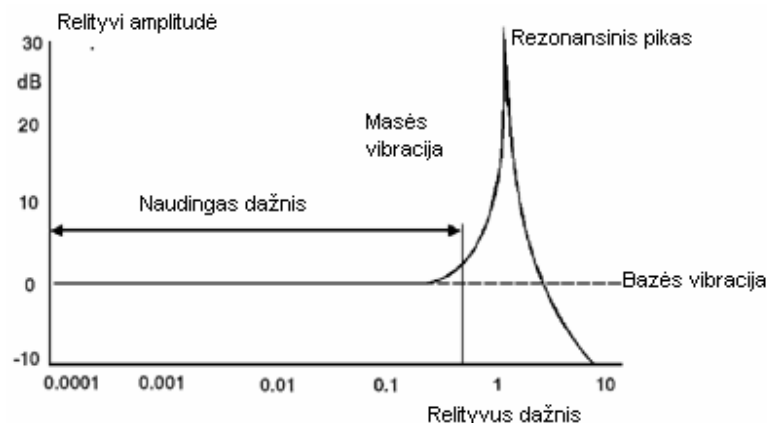
#### Pjezoelektriniai akselerometrai

Šie savaimė generuojantys prietaisai charakterizuojami išplėsta pastovaus dažnio juosta, didele tiesine amplitude bei puikiu patvarumu.



**1.1.2.1 pav. 3-jų kristalų pjezoelektrinis akselerometras. Tokia konfigūracija minimizuoja trumpalaikius šiluminius efektus ir bazės apkrovą.**

Pjezoelektrinės medžiagos, tokios kaip kvarcas ar rankų darbo keramika, turi savybę generuoti elektrinius signalus esant apkrovai. Akselerometre, pjezoelektriniai elementai atstoja spyruoklę su standumu  $k$ , ir jungia bazę su seismine mase. Sensorius veikia antrojo Niutono dėsnio pagrindu:  $F = ma$ . Esant pagreičiui, seisminė masė slekia kristalą. Efektyvus matavimo dažnis (1.1.2.2 pav.) apibrėžiamas sensoriaus rezonansinio dažnio  $\omega$ .



**1.1.2.2 pav. Naudinga dažnių juosta tipiniam pjezoelektriniam akselerometrui logaritminiam grafike.**

Rezonansinis dažnis:

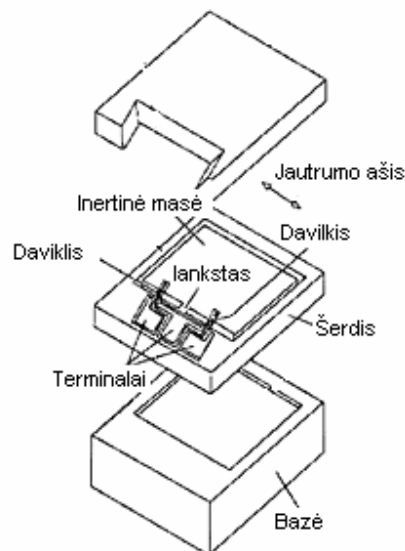
$$\omega = \sqrt{\frac{k}{m}} \quad (1.1.2.1)$$

Yra dvi pjezoelektrinių akselerometrų kategorijos (remiantis veikimo tipu). IEPE (integral electronic piezoelectric) tipas turi savyje integruotą signalo poveikio mechanizmą, kuris aukštos varžos signalą paverčia lengvai matuojama žemos varžos įtampa, kuri gali būti perduodama laidais į bet kokią įtampos matavimo prietaisą. *Charge-output* tipo prietaisai išduoda tik paties kristalo generuojamus itin aukštos varžos signalus.

Vartojimo paprastumas, didelis tikslumas, plati dažnių juosta ir maža kaina – IEPE akselerometrų savybės, kuriomis remiantis būtent jie pasirenkami vibracijų bei šoko matavimams. Jų viršutinė darbinė temperatūra dažniausiai yra 250F (121C), bet specialūs variantai gali dirbti net iki 350F (175C).

### Pjezo pasipriešinimo akselerometrai

Vieno-kristalo silikonas yra anisotropinė (anisotropic) medžiaga, kurios atomai yra išsidėstę tinkleliu, turinčiu kelias simetrijos ašis. Bet kurios plokštumos orientacija silikone apibrėžiama Milerio (Miller) indeksais. Gamybos procesas, ištobulintas aštuntame dešimtmetyje, išnaudoja pažangą silikono apdirbime. Vieno kristalo silikonas yra auginamas, tada pjaustomas, poliruojamas ir valomas. Kontroliuojama naudingųjų priemaišų difuzija paviršiniame regione. Foto litografija graviruoja juostą nurodytose vietose. Foto pasipriešinimas yra pašalinamas. Įvairios izotropinės ir anisotropinės cheminės medžiagos naudojamos mechaninės mikrostruktūros formavimui. Silikono pjezo pasipriešinimo koeficientas kontroliuojamas taikomo spaudimo pasiskirstymo mikrostruktūroje bei priemaišų kiekio. Plona plėvelės metalizacija atstoja ir rišantįjį padą, ir tiltą elektriniams signalams tarp kristale suformuotų paviršių. Tuomet lapas dalinamas į atskirus gabalus, kurie sujungiami į daviklio dėžutę. Įtempimo matuoklis yra integruojamas kaip tiltas, leidžiantis išgauti signalą ir nesant judesiui. Kadangi įtempimo matuokliai yra integruojami tiesiai į sulenkimus, šie akselerometrai naudoja silikoną ir kaip įsitempanti, ir kaip signalą keičiantį elementą.



#### 1.1.2.3 pav. Pjezo pasipriešinimo akselerometro struktūra

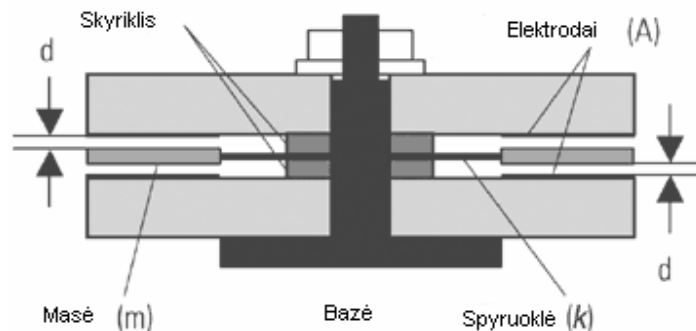
Dėl šių akselerometrų standumo rezonansinis dažnis yra didelis, todėl naudingo dažnio juosta yra plati. Kita itin trokštama savybė – mažas šių prietaisų dydis. Taip pat pasižymi geru teisiškumu ir



padidintu stabilumu. Esant geram kompensavimo mechanizmui, šio tipo akselerometrų darbinė temperatūra yra nuo -65F iki 250F.

### Talpiniai akselerometrai

Talpiniai akselerometrai veikimo principu panašūs į pjezo pasipriešinimo akselerometrus tuo, kad jie taip pat matuoja pasikeitimą tiltinėje grandinėje. Tačiau šiuo atveju matuojamas ne pasipriešinimo pokytis, bet elektrinės talpos pokytis. Sensorius susideda iš dviejų lygiagrečių plokštuminių talpos elementų. Vieno šios klasės tipo akselerometrai naudoja metalinę diafragmą kartu su aliuminėmis talpos plokštelėmis (1.1.2.4 pav.).



**1.1.2.4 pav. Talpinis akselerometras. Esant pagreičiui dėl masės atstumo pasikeitimo iki plokštelių, keičiasi talpa.**

Dvi plokštelės gaubia diafragmą, taip sukurdamos du talpumo elementus. Kuomet sensorius yra Žemės gravitacijos lauke ar įgyja pagreitį dėl vibracijos, masė veikiama jėgos  $F = ma$ . Kartu, tarp spyruoklės standumo ir jėgos galioja sąryšis:

$$X = \frac{F}{k} \quad (1.1.2.2)$$

Čia  $X$  – spyruoklės nuokrypis,  $k$  – standumo koeficientas.

Pastarasis nuokrypis iškraipo atstumą iki elektrodų. Tai turi tiesioginį efektą talpai:

$$C_2 = A_\epsilon \left( \frac{\epsilon}{d + X} \right) \quad (1.1.2.3)$$

Ir

$$C_2 = A_\epsilon \left( \frac{\epsilon}{d - X} \right) \quad (1.1.2.4)$$

Čia  $C$  – elementų talpa,  $A_\epsilon$  - spyruoklės standumas,  $\epsilon$  - oro dielektrinė skvarba,  $d$  – atstumas tarp masės ir elektrodo.

Talpiniai akselerometrai gali išmatuoti pagreitį nuo  $<2g$  iki šimtų  $g$  ar dažnius iki  $1\text{kHz}$  bei išverti šoką iki  $5000g$  ar netgi didesnę.

## 1.2 Skaitinio integravimo metodai

Uždavinys – apskaičiuoti integralą  $R = \int_a^b f(x)dx$ , čia  $a$  ir  $b$  – integravimo rėžiai, o  $f(x)$  – pointegralinė f-ja, pateikta reikšmių lentele. Visa skaitinio integravimo metodų esmė yra ta, kad pointegralinė f-ja  $f(x)$  keičiama aproksimuojančia funkcija  $F(x)$  ir laikoma, kad  $R \approx \int_a^b F(x)dx$ . Paprastai parenkama tokia aproksimuojančioji f-ja  $F(x)$ , kad integralas  $\int_a^b F(x)dx$  būtų lengvai integruojamas analiziškai.

### 1.2.1 Niutono ir Koteso kvadratūrinės formulės

Niutono ir Koteso kvadratūrinės formulės yra interpoliacinio tipo, kai pointegralinė f-ja  $f(x)$  keičiama interpoliaciniu polinomu, kurį nusako iš anksto fiksuoti vienodai nutolę vienas nuo kito taškai  $x_i$ . Tarkime, kad reikia apskaičiuoti integralą  $R = \int_a^b f(x)dx$ . Tada  $n$ -osios eilės Niutono ir Koteso kvadratūrinė formulė išvedama taip:

- 1) integravimo intervalas  $[a; b]$  taškais  $a = x_0, x_1, \dots, x_n = b$  dalijamas į  $n$  lygių dalių; čia  $x_i = x_0 + ih$ ,  $i = \overline{0, n}$ , o  $h = \frac{b-a}{n}$ ;
- 2) pointegralinė funkcija  $f(x)$  keičiama  $n$ -osios eilės interpoliaciniu polinomu  $F(x)$ , einančiu per taškus  $(x_i, y_i)$ ,  $i = \overline{0, n}$ ; čia  $y_i = f(x_i)$ ;
- 3) 
$$R = \int_a^b f(x)dx \approx \int_a^b F(x)dx = \sum_{i=0}^n w_i y_i \quad (1.2.1.1)$$

1.2.1.1 formulė turi tiksliai integruoti f-jas  $1, x, x^2, \dots, x^n$ . Tada simboliu  $m_k$  pažymėję  $\int_a^b x^k dx$  iš

1.2.1.1 formulės turime lygčių sistemą

$$w_0 x_0^i + w_1 x_1^i + \dots + w_n x_n^i = m_i, \quad i = \overline{0, n} \quad (1.2.1.2)$$

Išvesdami Niutono ir Koteso kvadratūrines formules, integralą pertvarkysime taip, kad svoriniai koeficientai  $w_i$  priklausytų tik nuo  $n$ .

Pavartoję keitinį  $z = x - a$ , gauname:

$$\int_a^b f(x)dx = \int_0^{b-a} f(z+a)dz = \int_0^{nh} f(z+a)dz.$$

Pažymėkime  $z = ht$ . Tada

$$\int_a^b f(x)dx = \int_0^{nh} f(z+a)dz = h \int_0^n f(a+ht)dt = \sum_{i=0}^n w_i f(a+ih) + K_n$$

čia  $K_n$  - kvadratūrinės formulės liekamasis narys.

Nedideliems  $n$  apskaičiuotus svorinius koeficientus  $w_i$  galima rasti literatūroje [4].

Verta paminėti, kad lyginės kvadratūrinės formulės liekamasis narys yra proporcingas  $h^{k+3}$  (čia  $k$  – lyginis skaičius), nelyginės –  $h^{l+2}$  (čia  $l$  – nelyginis skaičius). Taip pat, integravimo intervale taikant kvadratūrinę formulę kelis kartus, suminio liekamojo nario  $h$  laipsnis yra vienetu mažesnis.

### 1.2.2 Funkcijų, pateiktų reikšmių lentelė, integravimas

Tarkime, kad  $f(x)$  pateikta reikšmių lentelė  $(x_i, y_i)$ ; čia  $i = \overline{0, N}$ . Reikia apskaičiuoti  $\int_{x_0}^{x_N} f(x)dx$ .

Labai dažnai  $x_i$  reikšmės nesudaro aritmetinės progresijos, todėl Niutono ir Koteso kvadratūrinių formulių negalima taikyti, nes jos sudarytos vienodai nutolusiems vienas nuo kito taškams. Tokiu atveju, pointegralinę f-ją aproksimuojame interpoliaciniu kubiniu splineu  $g(x)$  ir laikome, kad

$$\int_{x_0}^{x_N} f(x)dx \approx \int_{x_0}^{x_N} g(x)dx.$$

Kaip pateikta literatūroje [4]:

$$\int_{x_0}^{x_N} g(x)dx = \sum_{i=1}^N \left( \frac{h_i}{2} (y_{i-1} + y_i) - \frac{h_i^3}{24} (m_{i-1} + m_i) \right) \quad (1.2.2.1)$$

čia  $y_i = f(x_i)$ ,  $m_i = g''(x_i)$ ,  $h_i = x_i - x_{i-1}$  kai  $i = \overline{0, N}$ .

Dažnai splineai užrašomi kaip tiesiniai B splineų deriniai [4]. Kai pointegralinė f-ja keičiama 3-iosios eilės interpoliaciniais splineais, užrašytas formule

$$g(x) = \sum_{k=-n}^{N-1} b_k B_n^k(x)$$

čia  $n = 3$ ,

turime:

$$R = \int_a^b f(x)dx \approx \int_a^b g(x)dx = \sum_{k=-n}^{N-1} \int_{x_k}^{x_{k+1}} B_n^k(x)dx.$$

Kadangi kvadratinėje formulėje yra dėmenų, rodančių, kad ne visas B splaino pagrindas priklauso intervalui  $[x_0; x_N]$ , tai remiantis literatūra [4] turime:

$$R = \int_a^b f(x)dx \approx \sum_{i=0}^{n-1} \sum_{k=i-n}^{-1} b_k \int_{x_i}^{x_{i+1}} B_n^k(x)dx + \frac{1}{n+1} \sum_{k=0}^{N-n-1} b_k (x_{k+n+1} - x_k) + \sum_{i=N-n}^{N-1} \sum_{k=N-n}^i b_k \quad (1.2.2.2)$$

$$\text{čia } \int_{x_i}^{x_{i+1}} B_n^k(x)dx = \frac{h}{3} (B_n^k(x_i) + 4B_n^k(x_i + h) + B_n^k(x_i + 2h)), \quad h = \frac{x_{i+1} - x_i}{2}.$$

### 1.3 Laisvas rotacijos centras

Tarkime, jog turime du judančius taškus A ir B tokius, kad A judėjimas yra apibrėžtas bet kokia f-ja, o taškas B sukasi apie tašką A. Jei

$$A = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \quad (1.3.1)$$

Tai taško B judėjimas apibrėžiamas:

$$B(t) = A(t) + \begin{pmatrix} R(t)\cos\alpha(t) \\ R(t)\sin\alpha(t) \end{pmatrix} \quad (1.3.2)$$

Čia  $R(t)$  yra taško B rotacijos spindulys apie tašką A laiko momentu  $t$ , o  $\alpha(t)$  – apibrėžia sukimosi dėsnį.

Pagal 1.3.2 išraišką nesunku pastebėti, kad jei išmestume iš taško B judėjimo taško A judėjimą, gautume, kad taškas B sukasi apie koordinatinių pradžių.

Dabar tarkime, jog taške B yra judanti koordinatinių plokštuma. Tuomet taško A koordinatės B sistemos atžvilgiu bus:

$$A'(t) = \begin{pmatrix} -R(t)\cos\alpha(t) \\ -R(t)\sin\alpha(t) \end{pmatrix} \quad (1.3.3)$$

1.3.3 formulė taip pat reiškia rotaciją apie koordinatinių pradžių tašką, tik kitaip nei taškas B, taškas A sukasi priešinga kryptimi. Dabar, jei prie taško A sukimosi pridėtume taško B (kaip koordinatinių sistemos, apie kurios pradžių tašką sukasi taškas A) judėjimą, gautume:

$$A'(t) + B(t) = A(t) \quad (1.3.4)$$

Po šių samprotavimų peršasi tokia labai svarbi išvada: jei turime du griežtai susijusius taškus A ir B, kur taškas A yra laisvai judantis, o taškas B – laisvai apie A besisukantis taškas, tuomet nesvarbu, kuri iš taškų A,B traktuosime kaip laisvai bejudantį tašką, o kuri kaip besisukantį. Dar daugiau, šalia šių dviejų taškų pasirinkę griežtai susijusį tašką C, galime taškus A ir B traktuoti kaip besisukančius apie tašką C.

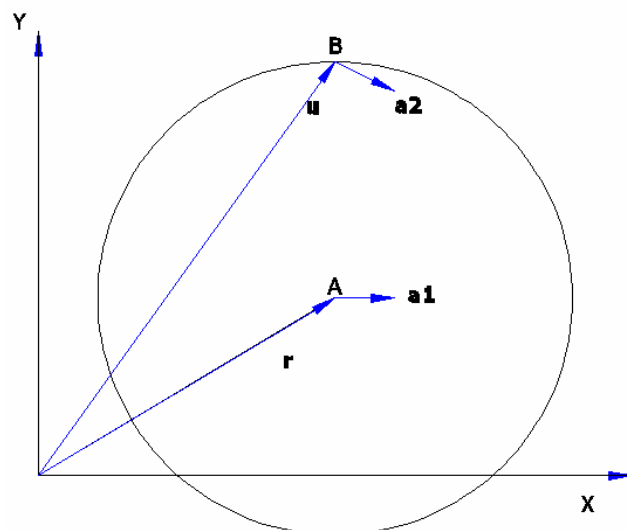
## 2 Tiriamoji dalis

### 2.1 Darbo tikslas

Užduotis ir tikslas – sudaryti tokį matematinį modelį, kuris leistų remiantis akselerometrų teikiamais duomenimis apie pagreitį, atkurti sistemos judesio trajektoriją, bei leistų užfiksuoti sistemos judėjimo metu atsiradusias deformacijas. Tuomet generuoti duomenis, juos analizuoti ir patikrinti modelio validumą.

### 2.2 Matematinis modelis, reikalavimai ir prielaidos

Remiantis 1.3 skyrelio išvada, jog bet kuris griežtai susijęs taškas gali būti rotacijos centru, galime rotacijos centru laikyti bet kurią sistemos akselerometrą. Kadangi turima galvoje tokios pačios dimensijos akselerometrai, kaip ir nagrinėjama erdvė, vadinasi, šiuo atveju mums reikalingas papildomas, griežtai susijęs su pirmuoju, taškas (akselerometras), kurio pagalba būtų galima žinoti sistemos padėtį globalios koordinatinių plokštumos atžvilgiu, nes tik tai leidžia teisingai interpretuoti akselerometrų išduodamus duomenis (čia turima galvoje, jog akselerometrai pagreičio vektorius matuoja savo vidinės koordinatinių sistemos atžvilgiu, todėl svarbu žinoti akselerometro orientaciją erdvėje, kad gautą pagreičio vektorius būtų galima transformuoti į globalios sistemos vektorius).



2.2.1 pav. Dviejų taškų (akselerometrų) sistema

Pastarasis modelis apibrėžiamas tokiomis diferencialinėmis lygtimis:

$$\frac{d^2 \vec{r}}{dt^2} = \vec{a1}(t) \quad (2.2.1)$$

$$\frac{d^2 \vec{u}}{dt^2} = \vec{a2}(t) \quad (2.2.2)$$

Kadangi tai yra paprastosios diferencialinės lygtys, jų sprendinys randamas du kartus integruojant dešiniojos lygties pusės funkciją. Atsižvelgę į problemos kontekstą, galime užrašyti:

$$\int_{t_0}^t \left( \int_{t_0}^z \vec{a}(u) du \right) dz = \int_{t_0}^t (\vec{v}(z) - \vec{v}_0) dz = \vec{r}(t) - \vec{r}_0 - \vec{v}_0(t - t_0) \quad (2.2.3)$$

čia  $\vec{v}(t) = \vec{r}'(t)$ ,  $\vec{a}(t) = \vec{v}'(t)$ ,  $\vec{r}_0 = \vec{r}(t_0)$ ,  $\vec{v}_0 = \vec{v}(t_0)$ .

Iš 2.2.3 galime užrašyti išraišką, kuri leidžia turint pagreičio vektoriaus funkciją gauti padėties vektorių laiko momentu  $t$ :

$$\vec{r}(t) = \int_{t_0}^t \left( \int_{t_0}^z \vec{a}(u) du \right) dz + \vec{r}_0 + \vec{v}_0(t - t_0) \quad (2.2.4)$$

Duotoji išraiška leidžia rasti judančio taško padėtį bet kuriuo laiko momentu ir kartu apibrėžia reikalavimus duomenims. Tarsime, kad modeliavimas ir duomenų nagrinėjimas visada prasideda laiko momentu  $t_0 = 0$ . Tuomet:

$$\vec{r}(t) = \int_0^t \left( \int_0^z \vec{a}(u) du \right) dz + \vec{r}_0 + \vec{v}_0 t \quad (2.2.5)$$

Matome, kad norėdami tiksliai rasti padėtį  $\vec{r}(t)$ , privalome žinoti pradinę padėtį  $\vec{r}_0$  bei pradinį greitį  $\vec{v}_0$ . Iš to, kas pasakyta, galime griežtai užrašyti modelį apibrėžiančias išraiškas:

$$\begin{aligned} \vec{r}(t) &= \int_0^t \left( \int_0^z \vec{a1}(u) du \right) dz + \vec{r}_0 + \vec{v}_{r0} t, \\ \vec{u}(t) &= \int_0^t \left( \int_0^z \vec{a2}(u) du \right) dz + \vec{u}_0 + \vec{v}_{u0} t \end{aligned} \quad (2.2.6)$$

čia  $\vec{r}_0$ ,  $\vec{u}_0$ ,  $\vec{v}_{r0}$  ir  $\vec{v}_{u0}$  yra žinomi.

## 2.3 Duomenų generavimas ir paruošimas

Darbe bus nagrinėjami duomenys, tariamai pamatuoti MEMSIC akselerometrais MXD2020G&M [5]. Tai dviejų ašių akselerometrai, kurių veikimas paremtas šilumos konvekcija. Prietaisai yra skaitmeniniai, galintys atlikti matavimus 100Hz ar 400Hz dažniu. Ir nors matavimo tikslumas priklauso

nuo daugelio faktorių, tokių kaip temperatūra ar matavimo netiesiškumas, tačiau šiame darbe į tai neatsižvelgsime tardami, kad matavimai atliekami idealiomis sąlygomis. Prietaisų tikslumą pagal jų veikimo parametrus apibrėžia ši lentelė [5]:

2.3.1 lentelė

**MXD2020G&M akcelerometrų tikslumas esant skirtingiems darbo parametrams**

Matavimo dažnis, Hz	Vidinio laikrodžio dažnis, MHz	Vidinio laikrodžio taktų skaičius vienam g	Tikslumas, mg
400	2.0	1000	1.0
400	1.0	500	2.0
400	0.5	250	4.0
100	2.0	4000	0.25
100	1.0	2000	0.5
100	0.5	1000	1.0

Remiantis pateikta lentele sudarytas akcelerometrų imitatorius, kuris naudojamas sugeneruotiems pagreičiams „pamatuoti“.

Pagreičiams generuoti naudojamos dviejų funkcijų išvestinės. Pirmoji f-ja apibrėžia taško A judėjimo trajektoriją, antroji – taško B rotaciją apie tašką A. Darbe bus naudojama po tris funkcijas kiekvienam taškui tokiu būdu gaunant devynias sistemos judesio trajektorijas. Taškui A naudojamos šios funkcijos (padėties, pagreičio):

$$\mathbf{A1:} \quad \begin{cases} A_x(t) = 0.1t^{2.1} \\ A_y(t) = \sin(t) \end{cases}, \quad \begin{cases} A_x''(t) = 0.231t^{0.1} \\ A_y''(t) = -\sin(t) \end{cases} \quad (2.3.1)$$

$$\mathbf{A2:} \quad \begin{cases} A_x(t) = \frac{1}{2}\sin(t) + \frac{1}{2}\cos(2t) \\ A_y(t) = \frac{1}{2}\cos(t) + \frac{1}{2}\cos(2t) \end{cases}, \quad \begin{cases} A_x''(t) = -\frac{1}{2}\sin(t) - 2\cos(2t) \\ A_y''(t) = -\frac{1}{2}\cos(t) - 2\cos(2t) \end{cases} \quad (2.3.2)$$

$$\mathbf{A3:} \quad \begin{cases} A_x(t) = \cos(t) \\ A_y(t) = \sin(t) \end{cases}, \quad \begin{cases} A_x''(t) = -\cos(t) \\ A_y''(t) = -\sin(t) \end{cases} \quad (2.3.3)$$

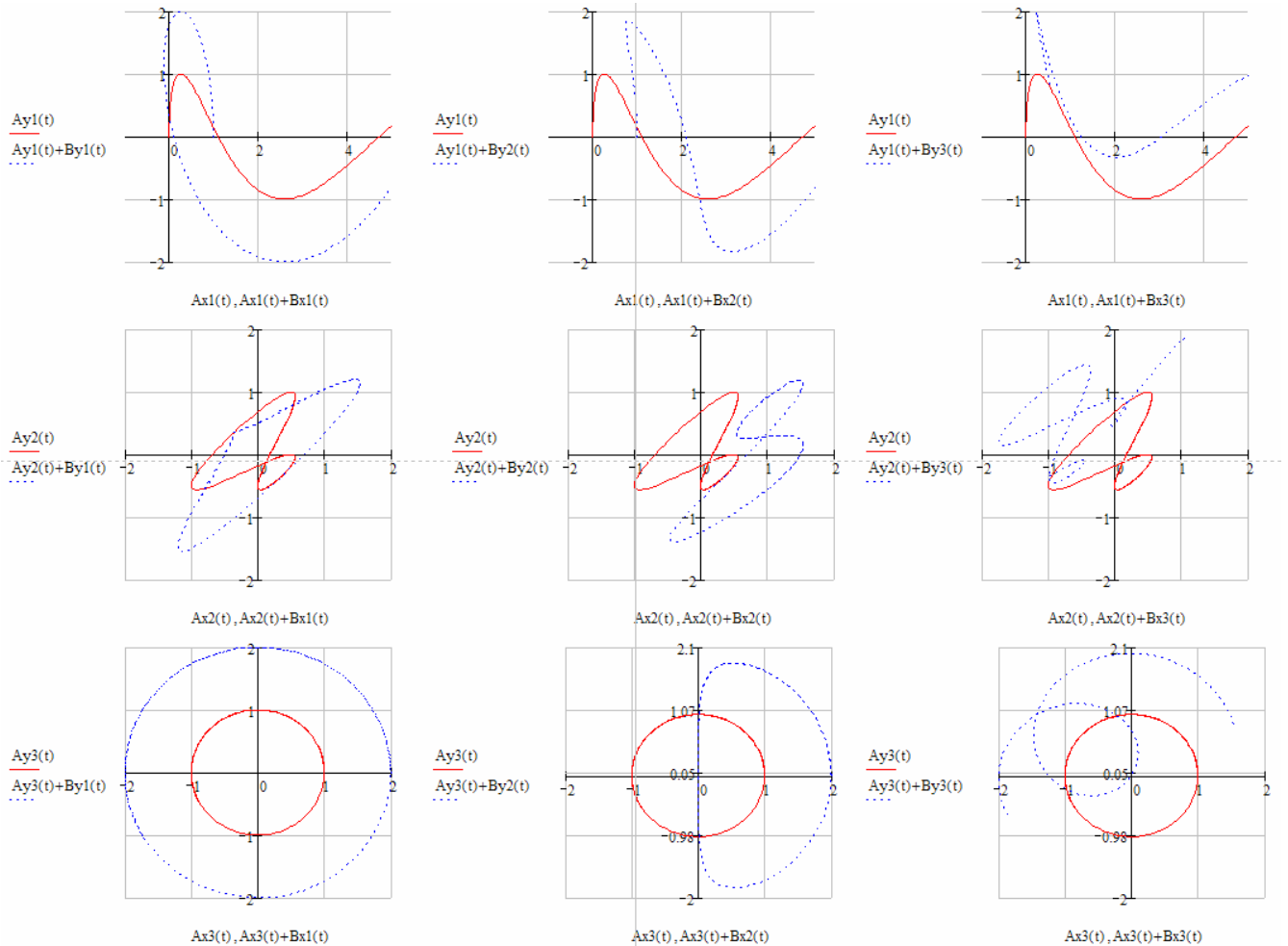
Taškui B naudojamos šios funkcijos:

$$\mathbf{B1:} \quad \begin{cases} B_x(t) = \cos(t) \\ B_y(t) = \sin(t) \end{cases}, \quad \begin{cases} B_x''(t) = -\cos(t) \\ B_y''(t) = -\sin(t) \end{cases} \quad (2.3.4)$$

**B2:** 
$$\begin{cases} B_x(t) = \cos(\sin(t)) \\ B_y(t) = \sin(\sin(t)) \end{cases}, \begin{cases} B_x''(t) = -\cos(\sin(t))\cos^2(t) + \sin(\sin(t))\sin(t) \\ B_y''(t) = -\sin(\sin(t))\cos^2(t) - \cos(\sin(t))\sin(t) \end{cases} \quad (2.3.5)$$

**B3:** 
$$\begin{cases} B_x(t) = \cos(\sqrt{t+1}) \\ B_y(t) = \sin(\sqrt{t+1}) \end{cases}, \begin{cases} B_x''(t) = -\frac{1}{4} \frac{\sin(\sqrt{t+1})}{t+1} - \frac{1}{4} \frac{\cos(\sqrt{t+1})}{(t+1)^{3/2}} \\ B_y''(t) = -\frac{1}{4} \frac{\cos(\sqrt{t+1})}{t+1} + \frac{1}{4} \frac{\sin(\sqrt{t+1})}{(t+1)^{3/2}} \end{cases} \quad (2.3.6)$$

Trajektorijos, kurias apibrėžia duotosios funkcijos, parodytos žemiau pateiktame paveiksle.



**2.3.1 pav. Darbe nagrinėjamos trajektorijos. Ax1(t) ir Ay1(t) atitinka 2.3.1 formulę, Ax2(t) ir Ay2(t) - 2.3.2, Ax3(t) ir Ay3(t) - 2.3.3, Bx1(t) ir By1(t) - 2.3.4, Bx2(t) ir By2(t) - 2.3.5, Bx3(t) ir By3(t) - 2.3.6.**

Kadangi darbe nagrinėjamos ir deformacijos, joms „sukurti“ pasirinktos šios funkcijos:

$$\Delta R(t) = t^3 \quad (2.3.7)$$

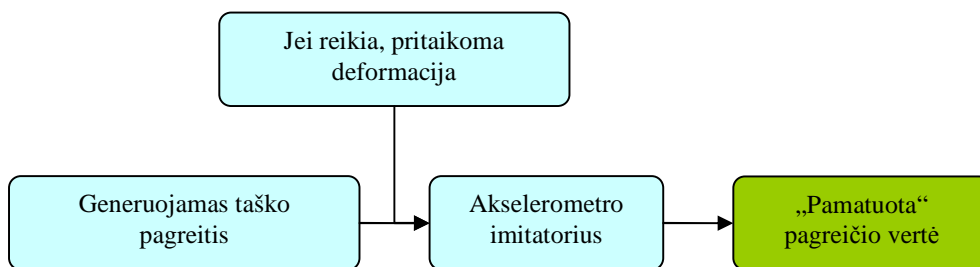
$$\Delta R(t) = t \quad (2.3.8)$$

$$\Delta R(t) = \sqrt{t} \quad (2.3.9)$$



Deformacijos funkcijos skirtos keisti taško B rotacijos spinduliui. Pasirinktu deformacijos pradžios laiko momentu  $t$  taško B rotacijos pagreičio dedamoji keičiama remiantis viena iš išvardintų funkcijų pasirinktą laiko tarpą. Tokių būdu gaunamos pagreičio vertės, iškraipančios judesio trajektoriją – įvyksta deformacija.

Bendra duomenų generavimo schema pateikta žemiau esančiame paveiksle.



2.3.2 pav. Duomenų generavimo schema

## 2.4 Integravimo metodo pasirinkimas

Kadangi norėdami rasti padėtį turime integruoti, iškyla skaitinio integravimo metodo pasirinkimo problema. Palyginsime tris skaitinio integravimo metodus: Niutono ir Koteso formules, pointegralinės funkcijos keitimą kubiniu interpoliaciniu splainu ir tiesiniu kubinių B splainų dariniu. Palyginimui imsime tris funkcijas:

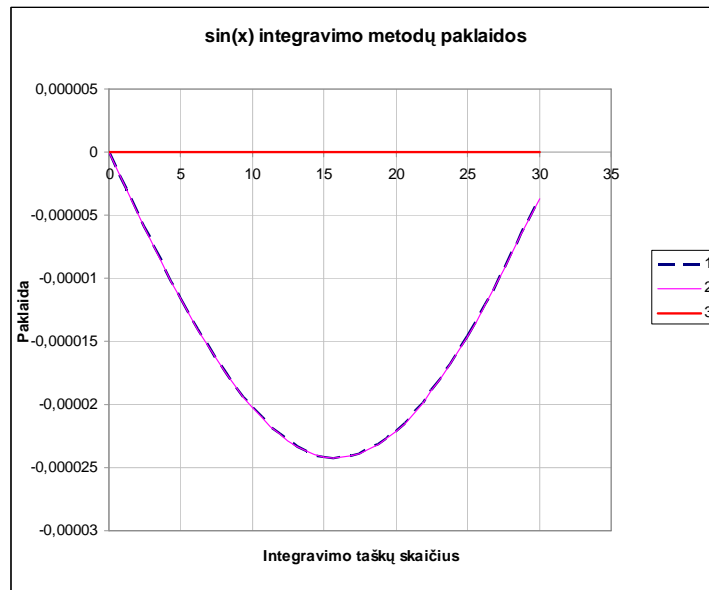
$$y(x) = \sin(x),$$

$$y(x) = \sin(x) + \sin(2x),$$

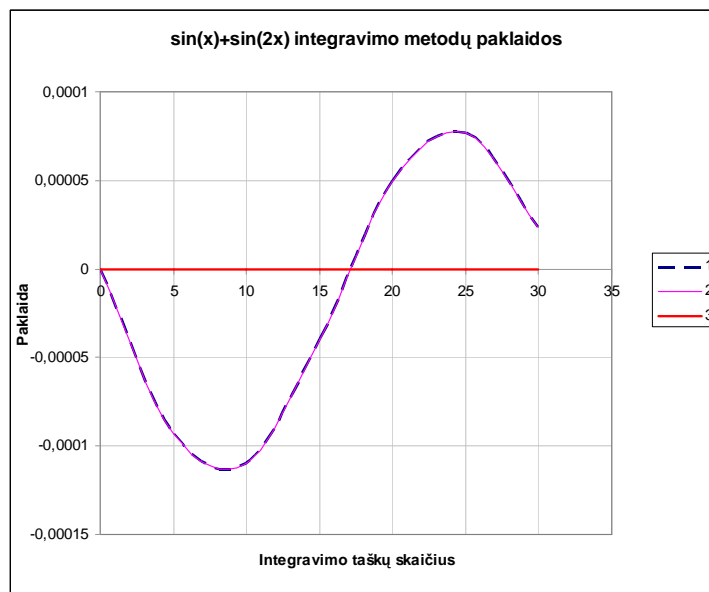
$$y(x) = \sqrt{x},$$

čia  $x \geq 0$ .

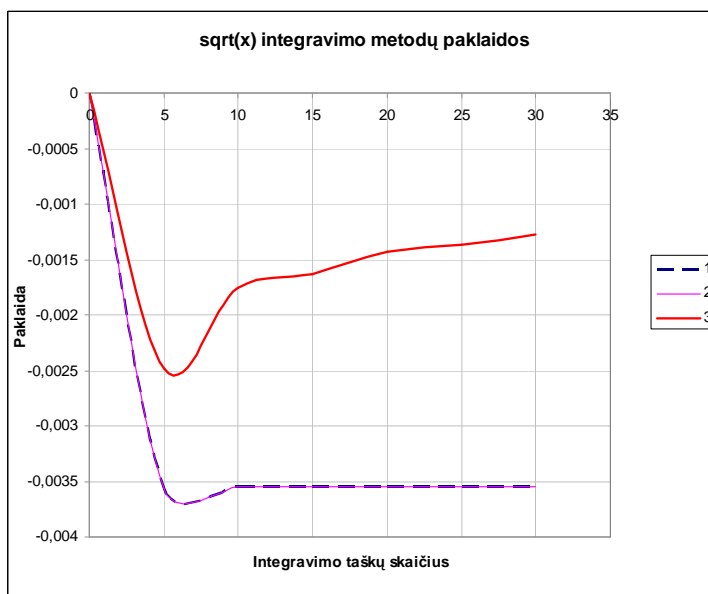
Žemiau pateikti trys grafikai (po vieną kiekvienai f-jai), rodantys, kaip kinta paklaidos ilgėjant integravimo intervalui, kai integravimui imami taškai, nutolę vienas nuo kito atstumu 0,1.



**2.4.1 pav. Funkcijos  $y(x) = \sin(x)$  integravimo paklaidos. 1 – naudojant kubinį interpoliacinį splineą, 2 – naudojant tiesinį kubinių B splineų darinį, 3 – remiantis Niutono ir Koteso formulėmis.**



**2.4.2 pav. Funkcijos  $y(x) = \sin(x) + \sin(2x)$  integravimo paklaidos. 1 – naudojant kubinį interpoliacinį splineą, 2 – naudojant tiesinį kubinių B splineų darinį, 3 – remiantis Niutono ir Koteso formulėmis.**



**2.4.3 pav. Funkcijos  $y(x) = \sqrt{x}$  integravimo paklaidos. 1 – naudojant kubinį interpoliacinį splainą, 2 – naudojant tiesinį kubinių B splainų darinį, 3 – remiantis Niutono ir Koteso formulėmis.**

Integravimas remiantis Niutono ir Koteso formulėmis akivaizdžiai skaičiuoja tiksliau, todėl šis metodas ir bus toliau naudojamas. Kaip kinta paklaidų reikšmės didinant integruojamą intervalą, pateikiamos 2.4.1 lentelėje.

**2.4.1 lentelė**

### Metodų paklaidos

Funkcija	$y(x) = \sin(x)$			$y(x) = \sin(x) + \sin(2x)$			$y(x) = \sqrt{x}$		
	Metodas			Metodas			Metodas		
Taškų skaičius	1	2	3	1	2	3	1	2	3
0	0,000E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00	0,000E+00
5	-1,159E-05	-1,159E-05	-5,611E-10	-9,358E-05	-9,358E-05	-6,963E-08	-3,564E-03	-3,564E-03	-2,485E-03
10	-2,032E-05	-2,032E-05	1,119E-15	-1,097E-04	-1,097E-04	1,399E-12	-3,547E-03	-3,547E-03	-1,755E-03
15	-2,415E-05	-2,415E-05	-1,189E-15	-4,001E-05	-4,001E-05	1,007E-14	-3,544E-03	-3,544E-03	-1,627E-03
20	-2,209E-05	-2,209E-05	3,394E-13	4,911E-05	4,911E-05	2,471E-14	-3,543E-03	-3,543E-03	-1,429E-03
25	-1,466E-05	-1,466E-05	7,694E-13	7,711E-05	7,711E-05	2,789E-12	-3,543E-03	-3,543E-03	-1,373E-03
30	-3,675E-06	-3,675E-06	6,487E-11	2,326E-05	2,326E-05	-3,809E-10	-3,542E-03	-3,542E-03	-1,271E-03

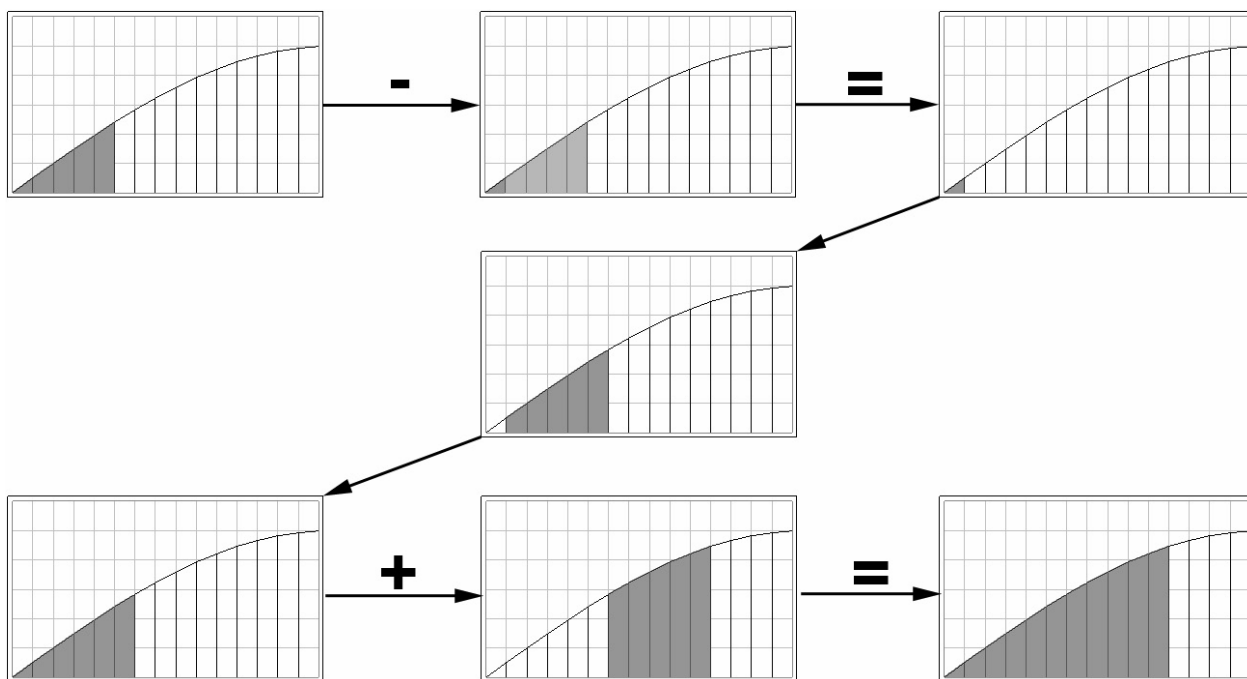
## 2.5 Integravimo strategija

Integruojant Niutono ir Koteso formulę pagalba, pointegralinė funkcija keičiama  $n$ -ojo laipsnio polinomu, kur  $n$  – integruojamų intervaliukų skaičius. Tokiu būdu iškyla ilgo integruojamo intervalo skaidymo į mažesnius intervalus problema. Reikia nustatyti, koks skaidymas reikalingas norint gauti mažiausias integravimo paklaidas. Taip pat, neužtenka vien tik skaidyti ilgo intervalo į trumpesnius, nes

tokiu būdu negaunamos jau suintegruotos funkcijos reikšmės kiekviename duomenų taške, būtinos antram integravimo žingsniui. Reikalinga tokia integravimo strategija, kuri užtikrintų ir didžiausią tikslumą, ir suintegruotos vieną (antram integravimo žingsniui) ir du kartus (deformacijoms fiksuoti) funkcijos reikšmės kiekviename duomenų taške. Abiems šioms sąlygoms patenkinti renkamės persiklojančio integravimo strategiją, kuri iliustruota 2.5.1 paveiksle.

Vykdomi tokie žingsniai:

- 1) pasirenkamas aproksimuojančio polinomo (kuris yra ir intervaliukų skaičius) laipsnis  $n$ ,
- 2) pirmieji  $n$  taškų randami pridendant skirtuminę vieno intervaliuko reikšmę prie jau sukauptos sumos iki nagrinėjamo intervaliuko,
- 3) visi kiti taškai randami integruojant  $n$  ilgio intervalą ir jį pridendant prie jau surastos reikšmės, nutolusios per  $n$  intervalų priešinga, nei vyksta integravimas, kryptimi.



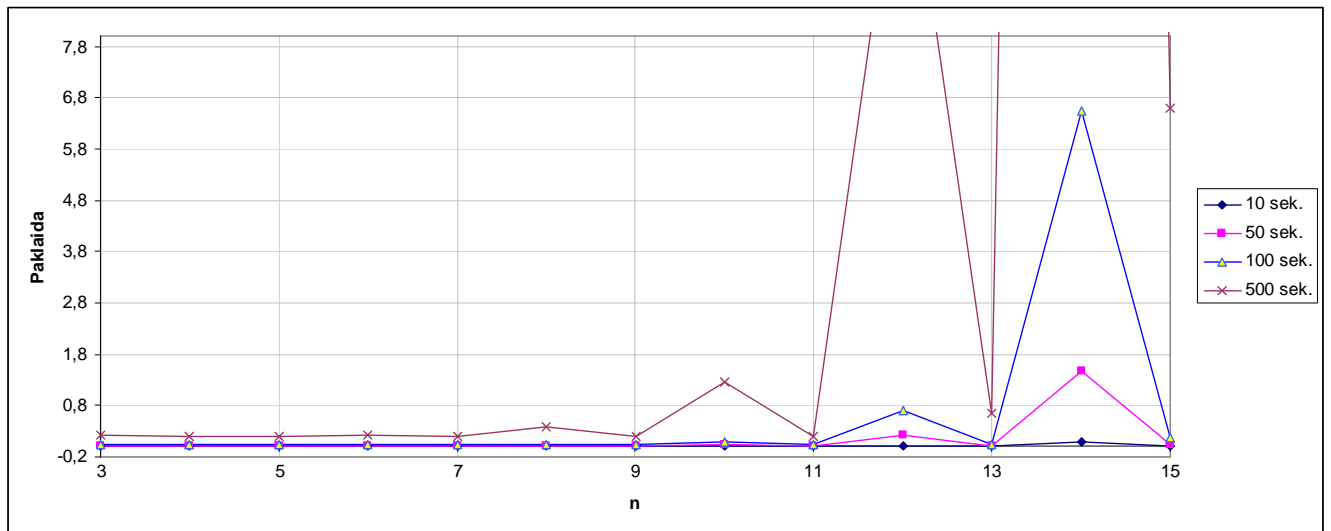
**2.5.1 pav. Integravimo strategija**

Pasirinkus integravimo metodiką, lieka ištirti paklaidų priklausomybę nuo šių faktorių: pasirinkto aproksimuojančio polinomo laipsnio  $n$ , integruojamo intervalo ilgio bei akselerometro darbo parametrų. Pirmiausia panagrinėkime paklaidų kitimą plačiam  $n$  diapazonui tyrimui pasirinkdami dvi trajektorijas:  $A1+B1$  (pirmoji trajektorija 2.3.1 paveiksle) ir  $A2+B2$  (ketvirtoji trajektorija 2.3.1 paveiksle). Tyrimo duomenys, kai akselerometro vidinio laikrodžio dažnis pasirinktas 2MHz, o matavimų dažnis – 100Hz, pateikti 2.5.1 lentelėje ir 2.5.2, 2.5.3 paveiksluose.

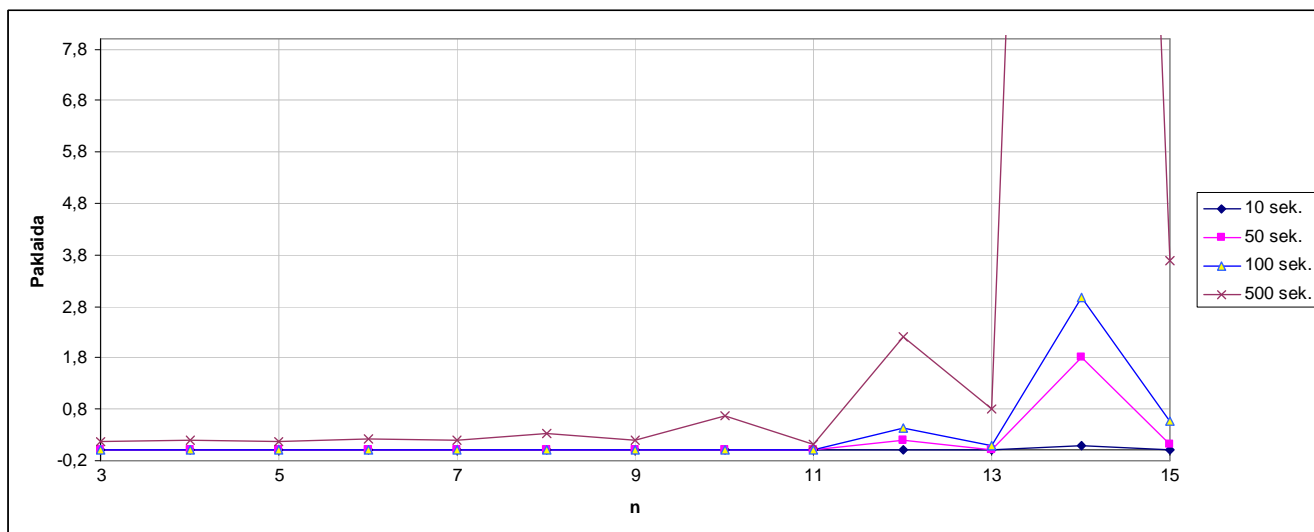
**2.5.1 lentelė**

**Paklaidų tyrimo rezultatai**

Trajekt.	n	Laiko intervalas, s												
		10			50			100			500			
		x	y	Pakl.	x	y	Pakl.	x	y	Pakl.	x	y	Pakl.	
A1+B1	3	-0,004252	0,000165	<b>0,004255</b>	-0,021738	0,003048	<b>0,021951</b>	-0,045755	0,009550	<b>0,046741</b>	-0,226239	-0,047817	<b>0,231237</b>	
	4	-0,003398	0,000160	<b>0,003402</b>	-0,017674	0,002287	<b>0,017821</b>	-0,038189	0,008034	<b>0,039025</b>	-0,185852	-0,082124	<b>0,203188</b>	
	5	-0,003656	0,000147	<b>0,003659</b>	-0,018758	0,003030	<b>0,019001</b>	-0,039945	0,009942	<b>0,041163</b>	-0,197379	-0,044091	<b>0,202243</b>	
	6	-0,003133	0,000157	<b>0,003137</b>	-0,016662	0,001343	<b>0,016716</b>	-0,036861	0,005230	<b>0,037230</b>	-0,176969	-0,128752	<b>0,218850</b>	
	7	-0,003357	0,000133	<b>0,003360</b>	-0,017147	0,003041	<b>0,017414</b>	-0,036571	0,008690	<b>0,037590</b>	-0,182772	-0,058147	<b>0,191798</b>	
	8	-0,002799	0,000281	<b>0,002813</b>	-0,015831	-0,002897	<b>0,016094</b>	-0,037593	-0,004095	<b>0,037815</b>	-0,167604	-0,344410	<b>0,383026</b>	
	9	-0,003182	0,000148	<b>0,003185</b>	-0,016980	0,003610	<b>0,017359</b>	-0,035317	0,011093	<b>0,037018</b>	-0,196672	-0,063764	<b>0,206750</b>	
	10	-0,000953	-0,001154	<b>0,001496</b>	-0,014910	-0,026509	<b>0,030414</b>	-0,061345	-0,061039	<b>0,086539</b>	-0,262289	-1,227520	<b>1,255229</b>	
	11	-0,002953	0,000033	<b>0,002954</b>	-0,015492	0,009160	<b>0,017997</b>	-0,032527	0,018909	<b>0,037624</b>	-0,201746	0,001543	<b>0,201752</b>	
	12	0,004236	-0,000059	<b>0,004237</b>	-0,047694	-0,216682	<b>0,221869</b>	-0,227108	-0,656737	<b>0,694897</b>	-1,144375	-12,496378	<b>12,548668</b>	
	13	-0,006193	0,000098	<b>0,006194</b>	0,000326	0,016938	<b>0,016941</b>	0,047085	-0,015576	<b>0,049595</b>	-0,635816	0,184161	<b>0,661950</b>	
	14	0,017736	-0,079718	<b>0,081668</b>	-0,036874	-1,462135	<b>1,462600</b>	-1,191511	-6,426581	<b>6,536103</b>	-8,923647	-92,709965	<b>93,138441</b>	
	15	0,002915	-0,005393	<b>0,006130</b>	0,038879	-0,009611	<b>0,040049</b>	-0,007114	0,171684	<b>0,171831</b>	-0,762488	6,546393	<b>6,590649</b>	
	A2+B2	3	0,000127	-0,000129	<b>0,000181</b>	0,001760	-0,000818	<b>0,001941</b>	0,004780	-0,008196	<b>0,009488</b>	0,092782	-0,156693	<b>0,182103</b>
		4	0,000128	-0,000156	<b>0,000202</b>	0,001329	-0,001260	<b>0,001831</b>	0,004435	-0,008737	<b>0,009798</b>	0,091253	-0,173433	<b>0,195975</b>
5		0,000136	-0,000129	<b>0,000187</b>	0,001877	-0,000779	<b>0,002033</b>	0,005224	-0,007838	<b>0,009420</b>	0,091593	-0,152517	<b>0,177907</b>	
6		0,000120	-0,000208	<b>0,000240</b>	0,000684	-0,001987	<b>0,002102</b>	0,003242	-0,009709	<b>0,010236</b>	0,090527	-0,198740	<b>0,218386</b>	
7		0,000153	-0,000150	<b>0,000214</b>	0,001495	-0,000831	<b>0,001711</b>	0,004826	-0,008775	<b>0,010015</b>	0,099363	-0,161040	<b>0,189227</b>	
8		0,000005	-0,000518	<b>0,000518</b>	-0,003263	-0,006571	<b>0,007337</b>	-0,004824	-0,015785	<b>0,016506</b>	0,055657	-0,319303	<b>0,324118</b>	
9		0,000060	-0,000176	<b>0,000186</b>	0,001648	-0,000644	<b>0,001769</b>	0,003714	-0,008241	<b>0,009039</b>	0,115187	-0,159811	<b>0,196997</b>	
10		0,000104	-0,001555	<b>0,001559</b>	-0,011837	-0,020058	<b>0,023291</b>	-0,013566	-0,015405	<b>0,020527</b>	0,007467	-0,675766	<b>0,675807</b>	
11		-0,000101	-0,000118	<b>0,000155</b>	0,003651	-0,003082	<b>0,004778</b>	0,008858	-0,010144	<b>0,013467</b>	0,082491	-0,071322	<b>0,109049</b>	
12		0,000786	-0,007075	<b>0,007118</b>	-0,098609	-0,168364	<b>0,195116</b>	-0,387003	-0,201066	<b>0,436118</b>	-1,162182	-1,883964	<b>2,213592</b>	
13		0,001086	0,000169	<b>0,001100</b>	0,011617	-0,008554	<b>0,014426</b>	0,029933	-0,072578	<b>0,078508</b>	-0,580747	0,562818	<b>0,808722</b>	
14		0,045147	-0,088824	<b>0,099639</b>	-1,611736	-0,836485	<b>1,815874</b>	-1,792697	-2,386359	<b>2,984706</b>	15,001944	-44,242178	<b>46,716471</b>	
15		0,009102	0,001868	<b>0,009292</b>	0,098377	-0,051819	<b>0,111191</b>	0,022278	0,569571	<b>0,570006</b>	1,836271	3,198598	<b>3,688214</b>	



2.5.2 pav. Paklaidų priklausomybė nuo n trajektorijai A1+B1



**2.5.3 pav. Paklaidų priklausomybė nuo n trajektorijai A2+B2**

Stebime metodo paklaidų šuolius kai  $n$  yra lyginis skaičius, todėl tolimesniuose tyrimuose šių  $n$  nebenagrinėsime. Analogiškus rezultatus gautume ir ištyrę kitas likusias trajektorijas. Remdamiesi gautais rezultatais tolimesniuose tyrinėjimuose  $n$  neimsime didesnio nei 13.

Mus taip pat domina, kaip kinta paklaidos renkantis skirtingus akcelerometro darbo parametrus. Nagrinėjant 100 s laiko intervalą, visoms devynioms trajektorijoms, gaunami rezultatai pateikiami 2.5.2 lentelėje.

**2.5.2 lentelė**

**Paklaidų priklausomybė nuo akcelerometro parametrų**

n = 5, t = 100s	Matavimų dažnis					
	100			400		
	Vid.laikr.dažnis, MHz			Vid.laikr.dažnis, MHz		
	2	1	0,5	2	1	0,5
Trajekt.	Pakl.	Pakl.	Pakl.	Pakl.	Pakl.	Pakl.
A1+B1	0,041163	0,041610	0,030256	0,010958	0,012791	0,009215
A1+B2	0,035462	0,043837	0,063082	0,032563	0,018595	0,141950
A1+B3	0,044403	0,019406	0,066842	0,067260	0,362762	0,223424
A2+B1	0,006539	0,009367	0,027510	0,013962	0,004039	0,017500
A2+B2	0,009420	0,006648	0,033137	0,028033	0,021673	0,151474
A2+B3	0,010664	0,057541	0,085648	0,073165	0,357247	0,229771
A3+B1	0,010536	0,008978	0,031067	0,012436	0,010423	0,006517
A3+B2	0,004598	0,012457	0,014551	0,020354	0,022220	0,129273
A3+B3	0,015060	0,047839	0,087774	0,074987	0,358726	0,211839

2.5.2 lentelėje nuspalvinti langeliai žymi paklaidas, esančias tarp mažiausios ir tarp du kartus didesnės paklaidų. Tokiu būdų tolimesniems tyrimams pasirenkame akcelerometrą, kuris matavimus atlieka 100Hz dažniu, ir kurio vidinio laikrodžio darbo dažnis yra 2 MHz.

**2.5.3 lentelė**

**Paklaidų priklausomybė nuo integruojamo intervaliuko dydžio**

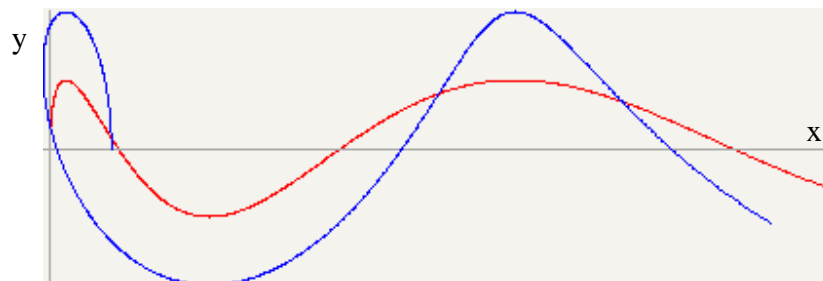
Trajekt.	n	Laiko intervalas, s					Laiko intervalas, s					Trajekt.
		10	50	100	500	1000	10	50	100	500	1000	
		Pakl.	Pakl.	Pakl.	Pakl.	Pakl.	Pakl.	Pakl.	Pakl.	Pakl.	Pakl.	
A1+B1	3	0,004255	0,021951	0,046741	0,231237	0,380788	0,000020	0,002027	0,010247	0,159514	2,955885	A2+B3
	4	0,003402	0,017821	0,039025	0,203188	0,294752	0,000055	0,002040	0,009883	0,154965	2,924562	
	5	0,003659	0,019001	0,041163	0,202243	0,326872	0,000019	0,002036	0,010664	0,156938	2,970757	
	6	0,003137	0,016716	0,037230	0,218850	0,304843	0,000119	0,002139	0,009279	0,148384	2,870007	
	7	0,003360	0,017414	0,037590	0,191798	0,296971	0,000058	0,002278	0,010059	0,161862	2,976245	
	8	0,002813	0,016094	0,037815	0,383026	0,691297	0,000475	0,003456	0,004804	0,168531	2,602216	
	9	0,003185	0,017359	0,037018	0,206750	0,280776	0,000037	0,002454	0,011313	0,161597	2,957983	
	10	0,001496	0,030414	0,086539	1,255229	2,587579	0,001872	0,004368	0,015890	0,450787	2,439764	
	11	0,002954	0,017997	0,037624	0,201752	0,424054	0,000221	0,001813	0,017981	0,162175	2,955202	
	13	0,006194	0,016941	0,049595	0,661950	2,194885	0,001663	0,018555	0,037362	0,452750	2,189927	
A1+B2	3	0,004316	0,021178	0,041410	0,258244	0,471726	0,000273	0,003064	0,010253	0,050519	0,103309	A3+B1
	4	0,003496	0,017529	0,034173	0,246693	0,445127	0,000295	0,002296	0,009286	0,084074	0,127641	
	5	0,003725	0,018217	0,035462	0,236228	0,420414	0,000280	0,003075	0,010536	0,046821	0,104974	
	6	0,003289	0,017115	0,033815	0,264112	0,492742	0,000341	0,001344	0,007855	0,130441	0,206864	
	7	0,003420	0,016946	0,032285	0,228144	0,405814	0,000251	0,003125	0,009168	0,062082	0,107000	
	8	0,003238	0,020253	0,041281	0,378838	0,840760	0,000607	0,002920	0,010241	0,345224	0,703057	
	9	0,003261	0,015866	0,030284	0,221543	0,405648	0,000227	0,003669	0,012282	0,083176	0,058872	
	10	0,003071	0,040932	0,082344	0,987657	2,482459	0,003510	0,027497	0,065714	1,229914	2,598984	
	11	0,003395	0,014222	0,027204	0,145074	0,707885	0,000577	0,009180	0,019471	0,075963	0,244363	
	13	0,005355	0,023221	0,051950	0,440549	2,191973	0,001810	0,019575	0,112711	0,863502	2,935213	
A1+B3	3	0,004422	0,024797	0,049789	0,126860	3,203210	0,000187	0,001175	0,004813	0,183257	0,302075	A3+B2
	4	0,003576	0,020729	0,042070	0,100872	3,139639	0,000184	0,001500	0,005569	0,205965	0,366408	
	5	0,003842	0,021914	0,044403	0,099446	3,194025	0,000205	0,001330	0,004598	0,180889	0,296814	
	6	0,003333	0,019737	0,040241	0,112221	3,079424	0,000181	0,002347	0,007469	0,238112	0,454553	
	7	0,003512	0,020530	0,041303	0,086940	3,185163	0,000192	0,001235	0,005419	0,188242	0,292596	
	8	0,003019	0,018806	0,038517	0,210842	2,815555	0,000145	0,007784	0,016726	0,381219	0,858382	
	9	0,003340	0,019622	0,039781	0,086274	3,170334	0,000198	0,000709	0,004863	0,190293	0,259365	
	10	0,002450	0,024075	0,060599	0,621935	2,069217	0,001271	0,028825	0,052763	1,034138	2,619567	
	11	0,003415	0,020064	0,042221	0,068687	3,045880	0,000452	0,001791	0,013854	0,065947	0,646726	
	13	0,005396	0,020995	0,033863	0,094377	3,414796	0,001083	0,032683	0,114742	0,633143	3,019235	
A2+B1	3	0,000237	0,003224	0,006087	0,055416	0,283902	0,000054	0,002861	0,014745	0,102158	3,090669	A3+B3
	4	0,000250	0,002712	0,004864	0,068005	0,305476	0,000073	0,002880	0,014426	0,109337	3,048976	
	5	0,000240	0,003304	0,006539	0,051725	0,277104	0,000060	0,002773	0,015060	0,101167	3,098459	
	6	0,000279	0,001951	0,002766	0,098018	0,359317	0,000103	0,003036	0,013779	0,121802	2,992832	
	7	0,000236	0,003269	0,005037	0,058924	0,279156	0,000066	0,002805	0,014482	0,102318	3,098731	
	8	0,000462	0,002554	0,008017	0,285071	0,816816	0,000318	0,003905	0,011810	0,215639	2,722129	
	9	0,000197	0,003537	0,007901	0,049660	0,248440	0,000028	0,003549	0,015816	0,086714	3,096445	
	10	0,003216	0,019144	0,037488	1,017329	2,567822	0,001944	0,008309	0,024370	0,592525	1,879206	
	11	0,000461	0,005007	0,017058	0,010751	0,221302	0,000180	0,006446	0,019817	0,080315	2,984796	
	13	0,000832	0,012871	0,075811	0,989325	2,006067	0,001879	0,021281	0,062599	0,301835	3,759827	
A2+B2	3	0,000181	0,001941	0,009488	0,182103	0,481391						
	4	0,000202	0,001831	0,009798	0,195975	0,523973						
	5	0,000187	0,002033	0,009420	0,177907	0,471778						
	6	0,000240	0,002102	0,010236	0,218386	0,595450						

7	0,000214	0,001711	0,010015	0,189227	0,463809
8	0,000518	0,007337	0,016506	0,324118	0,974537
9	0,000186	0,001769	0,009039	0,196997	0,422264
10	0,001559	0,023291	0,020527	0,675807	2,330550
11	0,000155	0,004778	0,013467	0,109049	0,656970
13	0,001100	0,014426	0,078508	0,808722	2,687434

2.5.3 lentelėje žalia spalva pažymėti mažiausios paklaidos langeliai, pilka – papildomų keturių mažiausių paklaidų langeliai. Tokiu būdu tai yra tarsi paklaidų žemėlapis. Remdamiesi 2.5.3 lentele pastebime, kad optimalus parametras intervaliuko ilgiui yra  $n = 9$ . Optimalus jis ta prasme, kad duoda mažiausias arba netoli mažiausios paklaidas daugumoje atveju. Reikia paminėti, kad visi tyrimai atlikti taškui B, nes jo dinamika yra didesnė (didesnė tikimybė atsirasti paklaidoms).

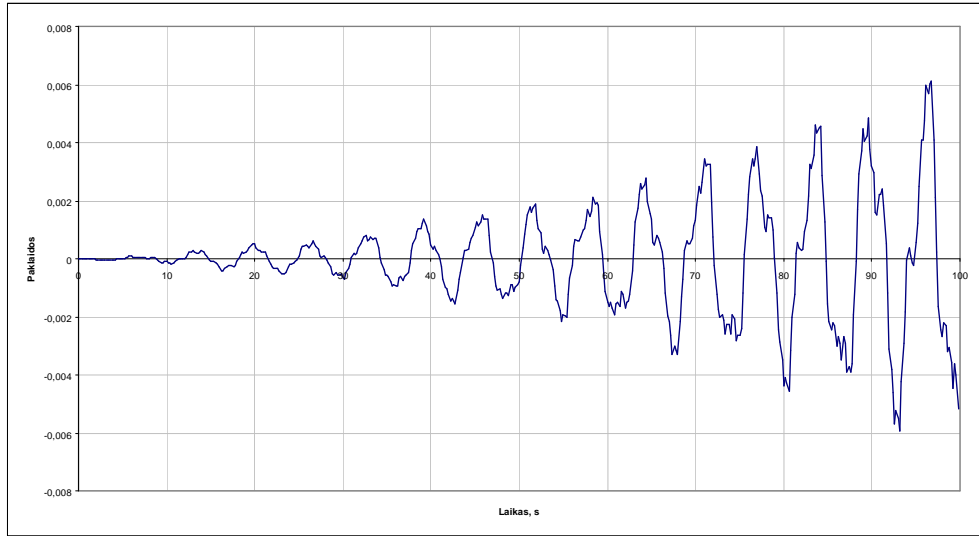
## 2.6 Trajektorijos atkūrimas. Paklaidos

Skyreliuose 2.4-2.5 pasirinkome integravimo metodą, aptarėme integravimo strategiją ir nustatėme optimaliausius darbo parametrus, t.y. pasirinkome akselerometrą su 2Mhz vidinio laikrodžio dažniu bei 100Hz matavimo dažniu, taip pat parinkome intervaliukų, kuriais skaidomas visas integravimo intervalas, ilgį  $n$ . Dabar bandysime iširti, kaip gerai nagrinėjamas modelis atkuria judesio trajektorijas. Kartu stebėsime ir paklaidų dinamiką. Atkurta trajektorija vaizdiniam palyginimui su pradine bei paklaidų grafikai pateikti žemiau esančiuose paveiksluose kiekvienai trajektorijai.

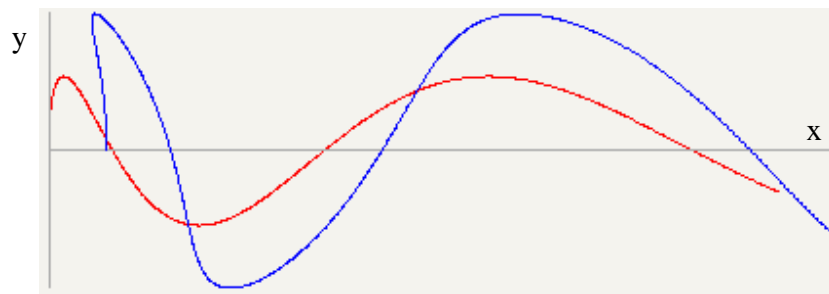


2.6.1 pav. Atkurta trajektorija A1+B1

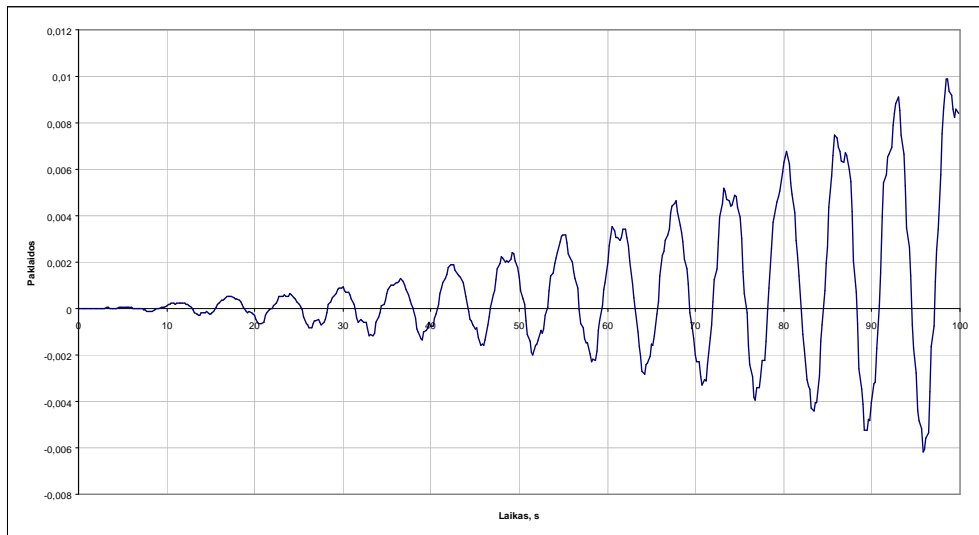




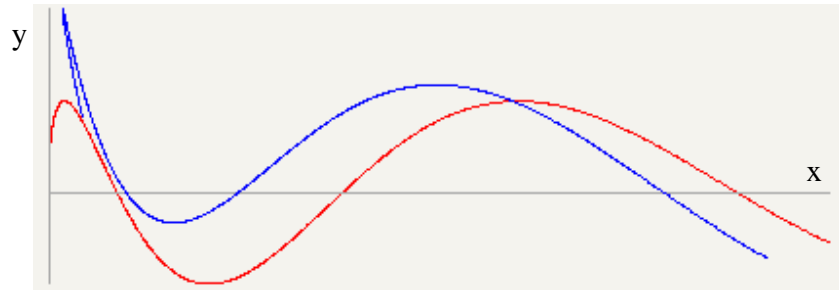
**2.6.2 pav. Paklaidos taškui B trajektorijoje A1+B1**



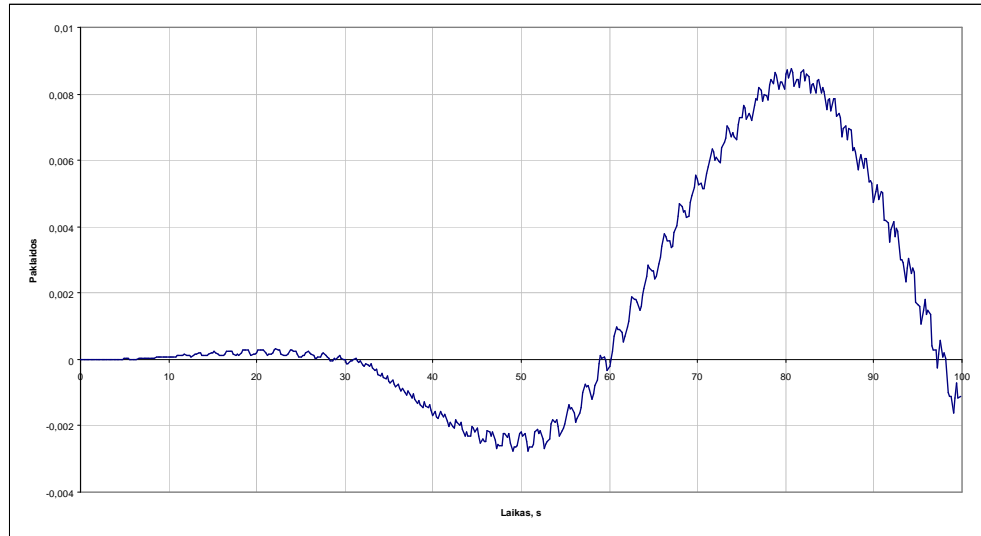
**2.6.3 pav. Atkurta trajektorija A1+B2**



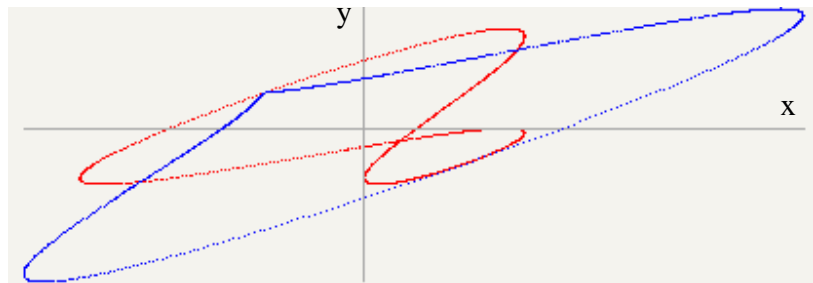
**2.6.4 pav. Paklaidos taškui B trajektorijoje A1+B2**



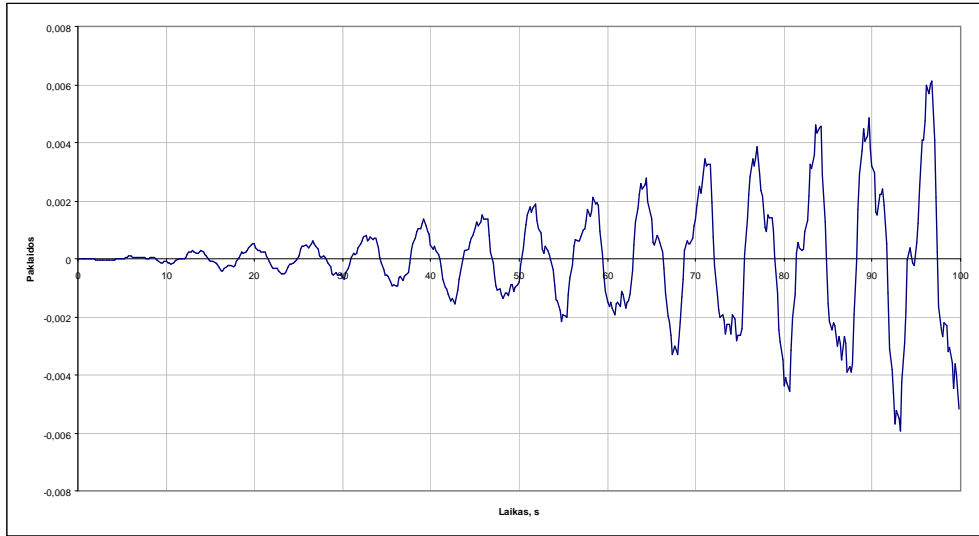
**2.6.5 pav. Atkurta trajektorija A1+B3**



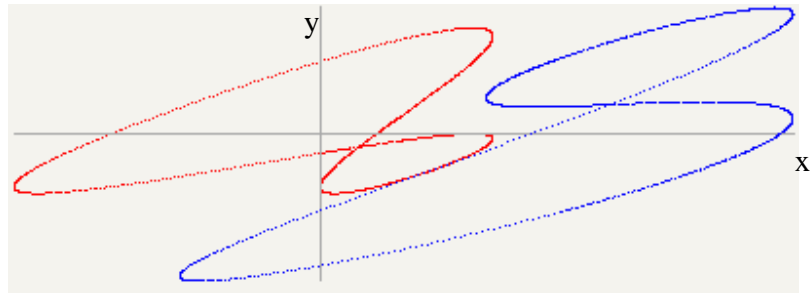
**2.6.6 pav. Paklaidos taškui B trajektorijoje A1+B3**



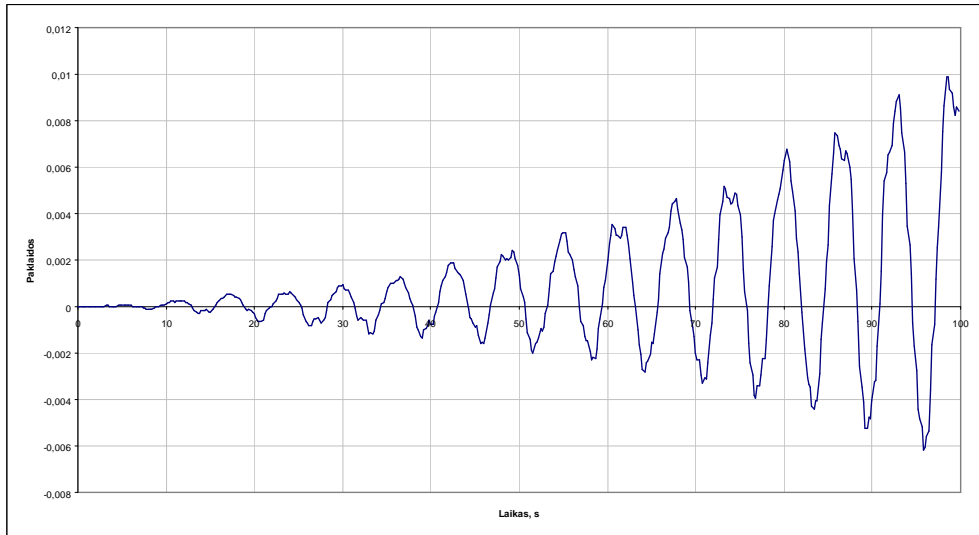
**2.6.7 pav. Atkurta trajektorija A2+B1**



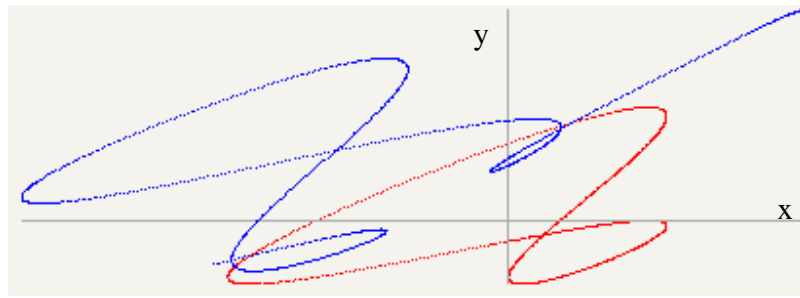
**2.6.8 pav. Paklaidos taškui B trajektorijoje A2+B1**



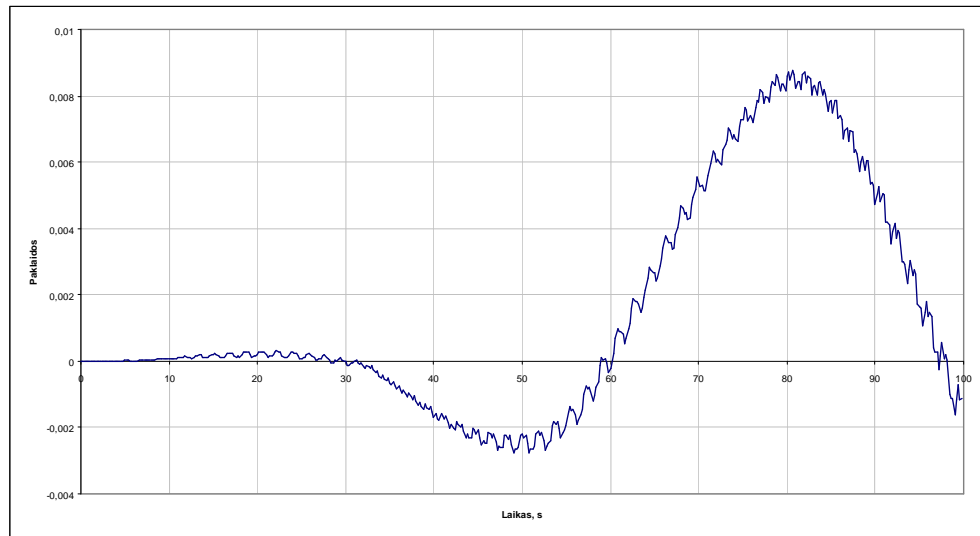
**2.6.9 pav. Atkurta trajektorija A2+B2**



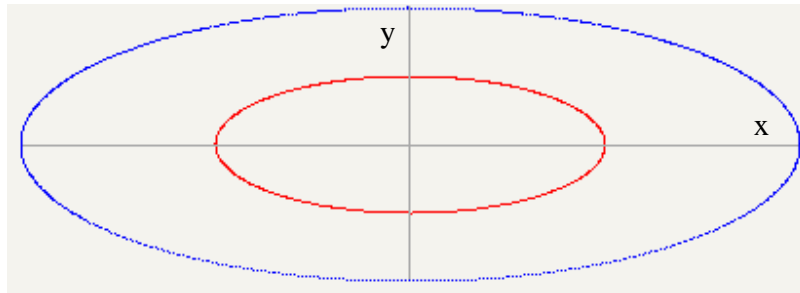
**2.6.10 pav. Paklaidos taškui B trajektorijoje A2+B2**



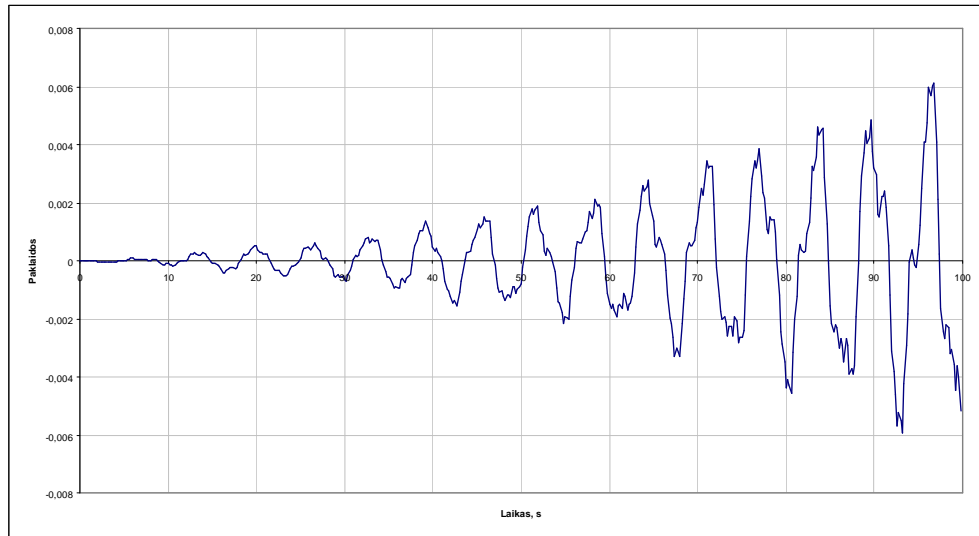
2.6.11 pav. Atkurta trajektorija A2+B3



2.6.12 pav. Paklaidos taškui B trajektorijoje A2+B2



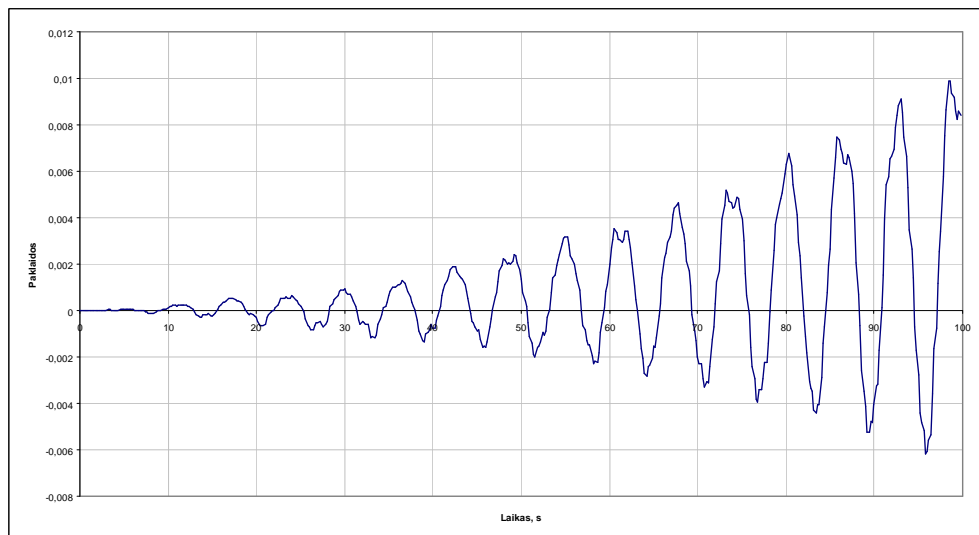
2.6.13 pav. Atkurta trajektorija A3+B1



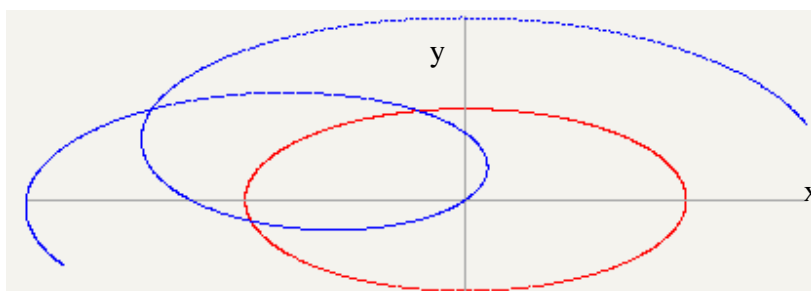
**2.6.14 pav. Paklaidos taškui B trajektorijoje A3+B1**



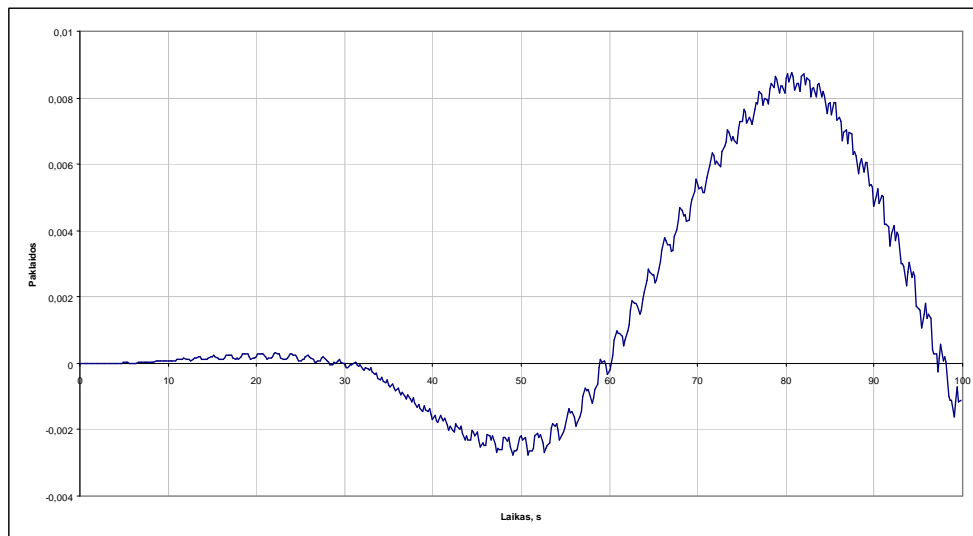
**2.6.15 pav. Atkurta trajektorija A3+B2**



**2.6.16 pav. Paklaidos taškui B trajektorijoje A3+B2**



2.6.17 pav. Atkurta trajektorija A3+B3



2.6.18 pav. Paklaidos taškui B trajektorijoje A3+B3

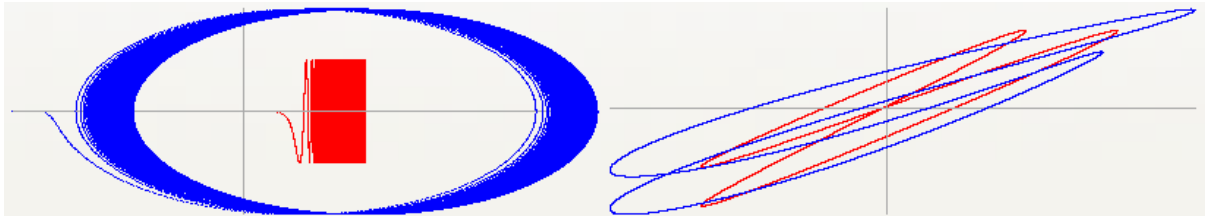
Kaip matome iš 2.6.1 - 2.6.18 paveikslų, atkuriamos trajektorijos atitinka pradines trajektorijas. Taip pat pastebime, jog paklaidų charakteris kas trečiai trajektorijai sutampa, todėl tolimesniuose paklaidų tyrimuose nagrinėsime tik pirmųjų trijų trajektorijų duomenis. Reikia paminėti, kad duotuosiuose grafikuose trajektorijų grafikai atkurti iki laiko momento  $t = 10$  s, o paklaidų grafikai pateikti iki laiko momento  $t = 100$  s. Skaitinės paklaidų reikšmės, kai judantys taškai stebimi vieną valandą, pateikiami 2.6.1 lentelėje.

2.6.1 lentelė

### Paklaidos praėjus vienai valandai

Tr.	Tšk.	Tikroji padėtis		Nueitas kelias	Paklaidos pagal koordinatę		Pakl.	Sant. pakl.
		x	y		x	y		
A1+B1	A	2939240,157524	-0,035909	2939245,38881	-1,211701	0,226174	1,232629	0,00004%
	B	2939241,199258	-0,071819	2939297,58842	-1,135012	0,452349	1,221832	0,00004%
A2+B2	A	0,300270	0,913835	3665,97114	3,086026	-2,316363	3,858638	0,10526%
	B	1,266122	0,654741	4794,40263	4,496898	-4,293760	6,217593	0,12968%
A3+B3	A	0,965045	-0,262084	3600,00000	-0,000292	0,226174	0,226175	0,00628%
	B	0,015205	-0,574820	3602,57960	-8,185241	33,972228	34,944391	0,96998%

Žiūrint į 2.6.1 lentelę kyla klausimas, kodėl antruoju ir trečiuoju atveju paklaidos yra gana didelės. Gali net pasirodyti, jog skaičiavimo metodika yra netinkama. Taip nėra. Tokios paklaidos grindžiamos pirmiausia prietaiso paklaidų dydžiu, antra – jei palygintume  $A1+B1$  ir tarkim  $A2+B2$  pagreičių kreives, pamatytume, jog pirmuoju atveju pagreičio kreivės neturi staigių lankų lyginant su antruoju atveju. Tuo galime įsitikinti 2.6.19 paveikslo pagalba.



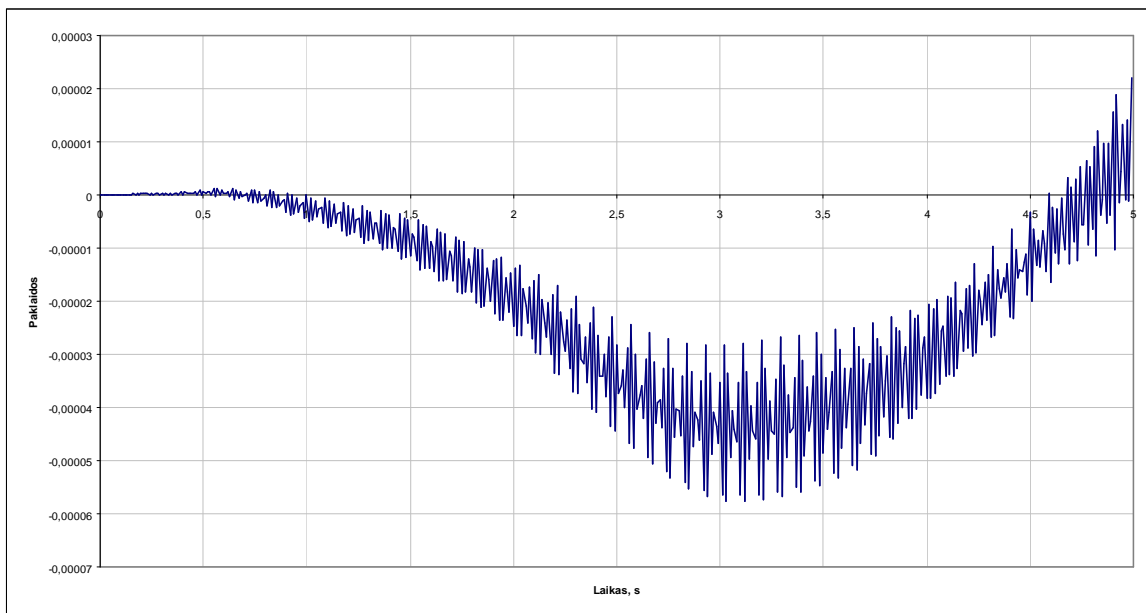
**2.6.19 pav. Pagreičių funkcijų grafikai trajektorijoms  $A1+B1$  ir  $A2+B2$**

Trajektorija  $A1+B1$  panaši į realaus pasaulio kūno judėjimą, nors ir nukelias atstumas šiandien mūsų technologiniam išsivystymui sunkiai pasiekiamas, o trajektorija  $A2+B2$  būtų artima kažkokio tipo vibracijai. Tokie pamąstymai leidžia daryti prielaidas apie akselerometrų sistemas, kuriose būtų naudojami pasirinkti akselerometrai, taikymo sritis.

## 2.7 Deformacijų fiksavimas

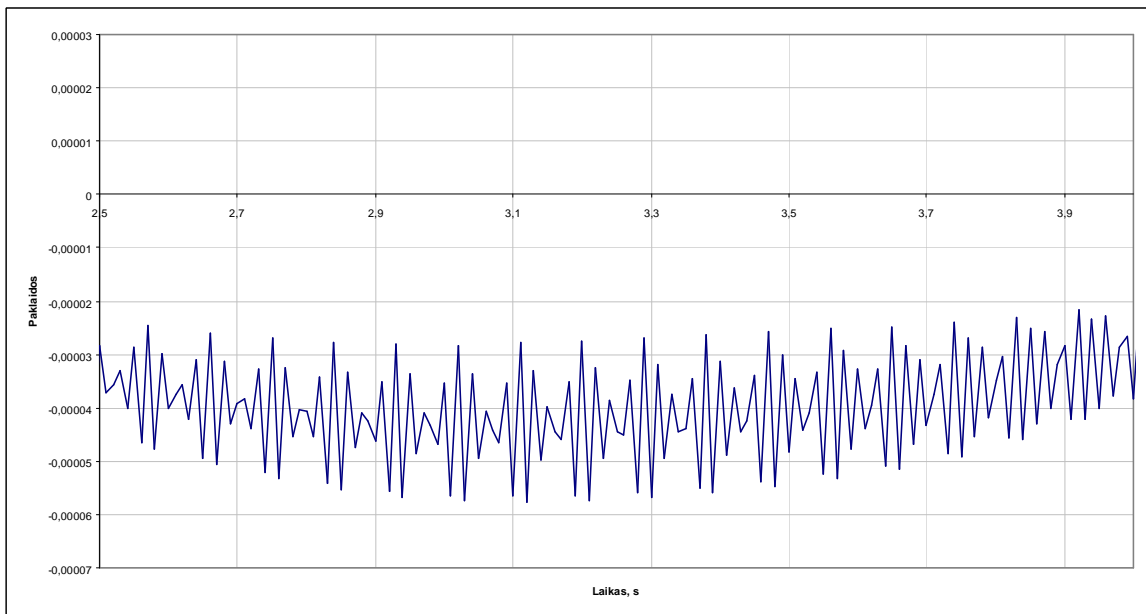
Tolimesnis uždavinys – aptikti deformacijos pradžią atkuriant judesį. Tam tikslui reikalingas metodas, leidžiantis tarsi pažinti, kur atstumo tarp taškų pokytis yra paklaidų priežastis, o kur – jau deformacijos priežastis. Mechanizmo išaiškinimui pasinaudokime  $A1+B1$ ,  $A2+B2$  ir  $A3+B3$  trajektorijomis, nes šios trajektorijos atitinka tris skirtingus nagrinėjamų trajektorijų paklaidų charakterius (2.6.1 – 2.6.18 paveikslai). Pirmiausia nagrinėkime trajektorijos  $A1+B1$  paklaidas, kurių grafikas pateiktas 2.6.2 paveiksle.

Matome, jog paklaidos yra periodinės su vis didėjančia amplitude. Be to, pastebime stiprų paklaidų svyravimą. Detaliau panagrinėkime intervalą iki  $t = 5$  s.



**2.7.1 pav. Paklaidų grafikas trajektorijai A1+B1 iki  $t = 5$  s**

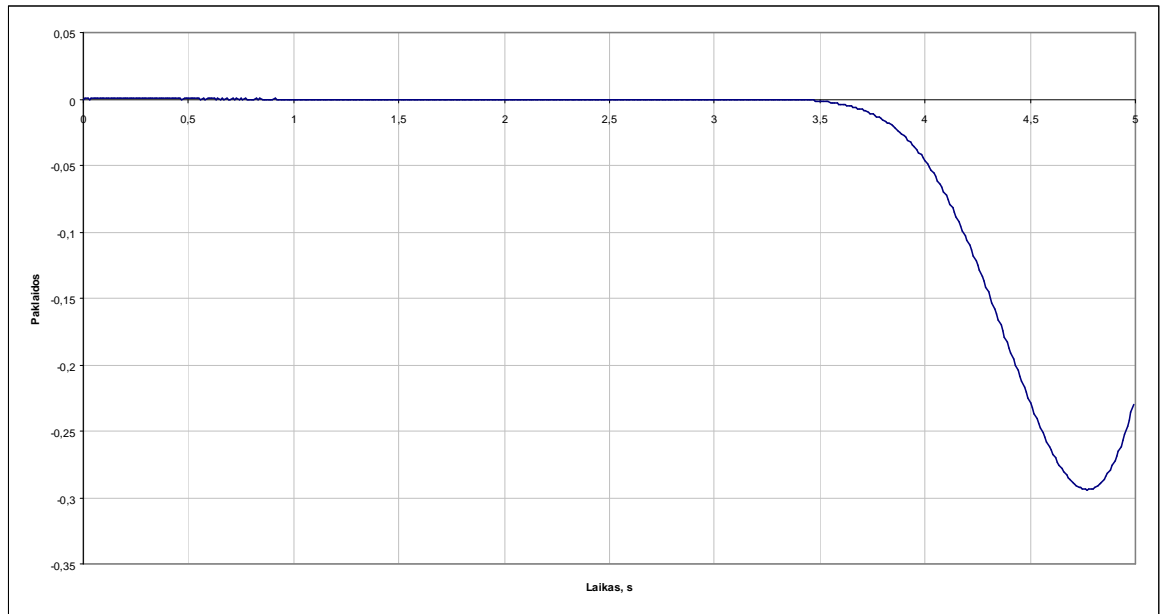
Kaip matyti iš 2.7.1 paveikslas, nors paklaidos ir turi globalų dėšningą kitimą, tačiau trumpame intervale labai intensyviai kinta (2.7.2 paveikslas).



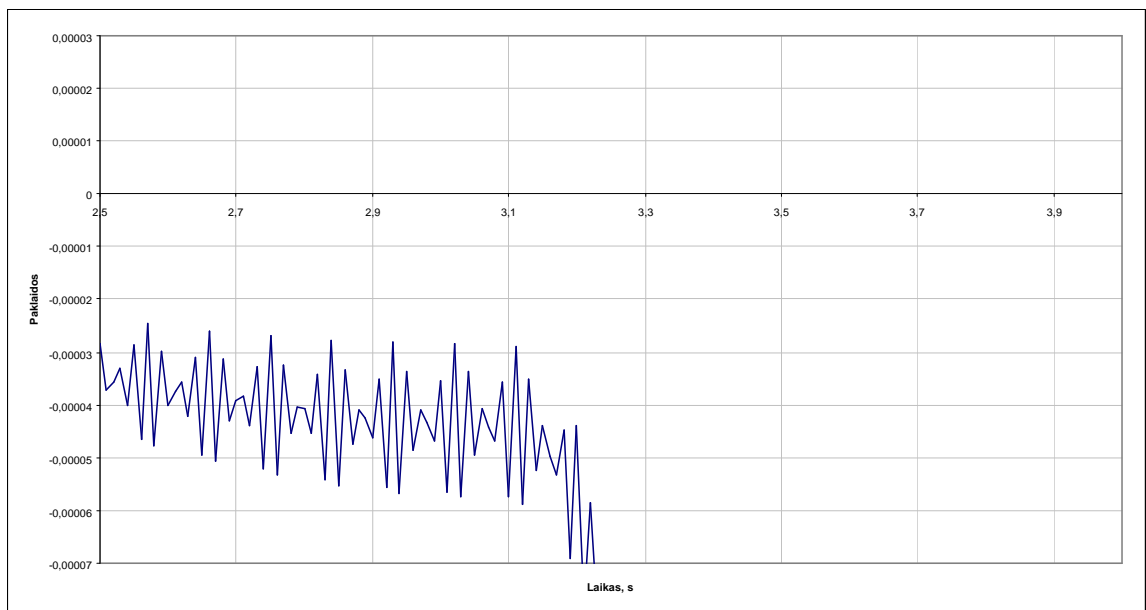
**2.7.2 pav. Paklaidų grafikas trajektorijai A1+B1  $t = [2,5;4]$**

Kaip pakis paklaidų grafikas, jei laiko momentu  $t = 3$  s įvesime deformaciją pagal 2.3.7 formulę?  
Atsakymas pateiktas 2.7.3 - 2.7.4 paveiksluose.





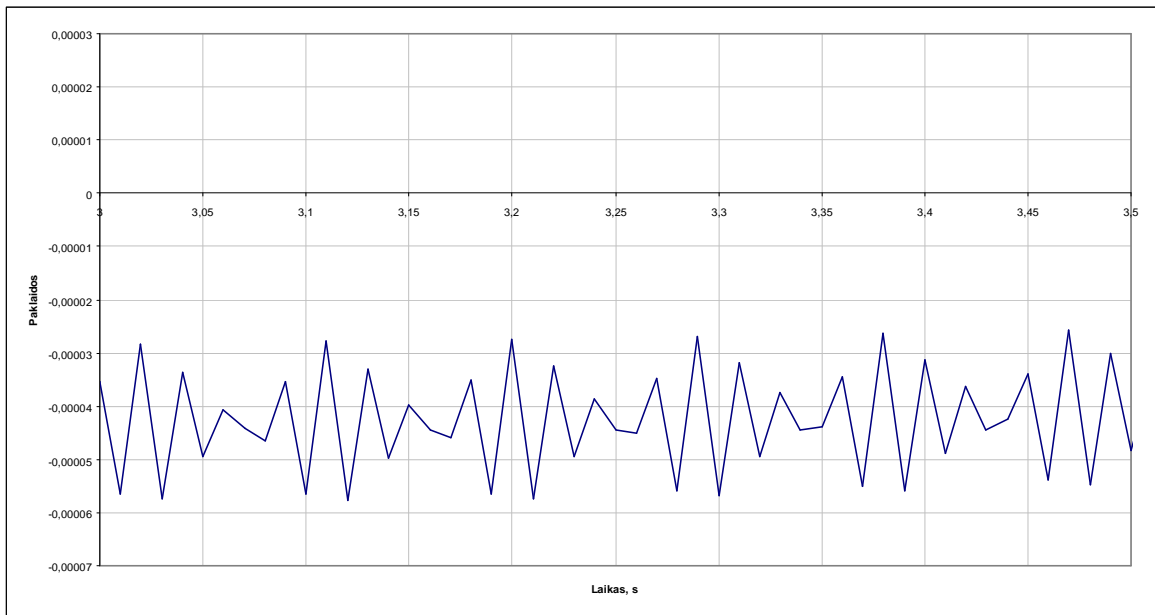
**2.7.3 pav. Paklaidų grafikas trajektorijai A1+B1 iki  $t = 5$  s deformacijai įvykus laiko momentu  $t = 3$  s**



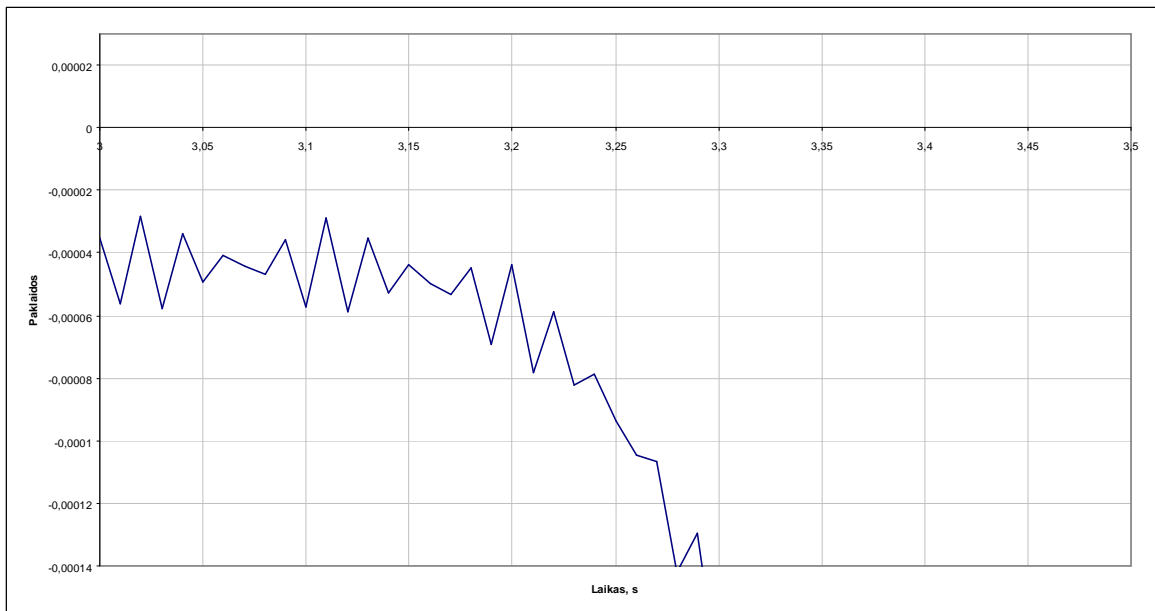
**2.7.4 pav. Paklaidų grafikas trajektorijai A1+B1  $t = [2,5;4]$  deformacijai įvykus laiko momentu  $t = 3$  s**

Lygindami 2.7.1 ir 2.7.3 paveikslus akivaizdžiai matome, kaip pakito paklaidų charakteris. Pažiūrėję į 2.7.4 paveikslą „iš akies“ galime daryti grubias išvadas apie deformacijos momento pradžia, tačiau puikiai suprantame, kad kompiuteriui reikalingas mechanizmas ir kriterijai, kuriais jis gali remtis. Be to, kompiuteris išvadas apie deformacijas turi daryti remdamasis kuo mažesniu duomenų kiekiu, t.y. mes negalime lyginti toli viena nuo kitos laiko prasme nutolusių paklaidų, nes tuomet nežinosime paklaidų

pokyčio priežasčių ir sprendimas apie deformacijas bus nepagrįstas. Sprendimas apie tai, ar įvyko deformacija, turi būti paremtas nagrinėjant vieno integruojamo intervaliuko, kurio ilgis, kaip pasirinkome 2.4 skyriuje, yra  $N = 10$ , t.y. sprendimas turi būti grindžiamas vos dešimčia taškų, arba kitaip tariant – laiko intervalu, kurio ilgis vos 0,1 sekundės. Palyginimui, kaip atrodo paklaidų grafikai su deformacija ir be jos, pateikti 2.7.5 ir 2.7.6 paveiksluose.



**2.7.5 pav. Paklaidų grafikas trajektorijai A1+B1  $t = [3;3,5]$**



**2.7.6 pav. Paklaidų grafikas trajektorijai A1+B1  $t = [3;3,5]$  deformacijai įvykus laiko momentu**

$$t = 3\text{s}$$

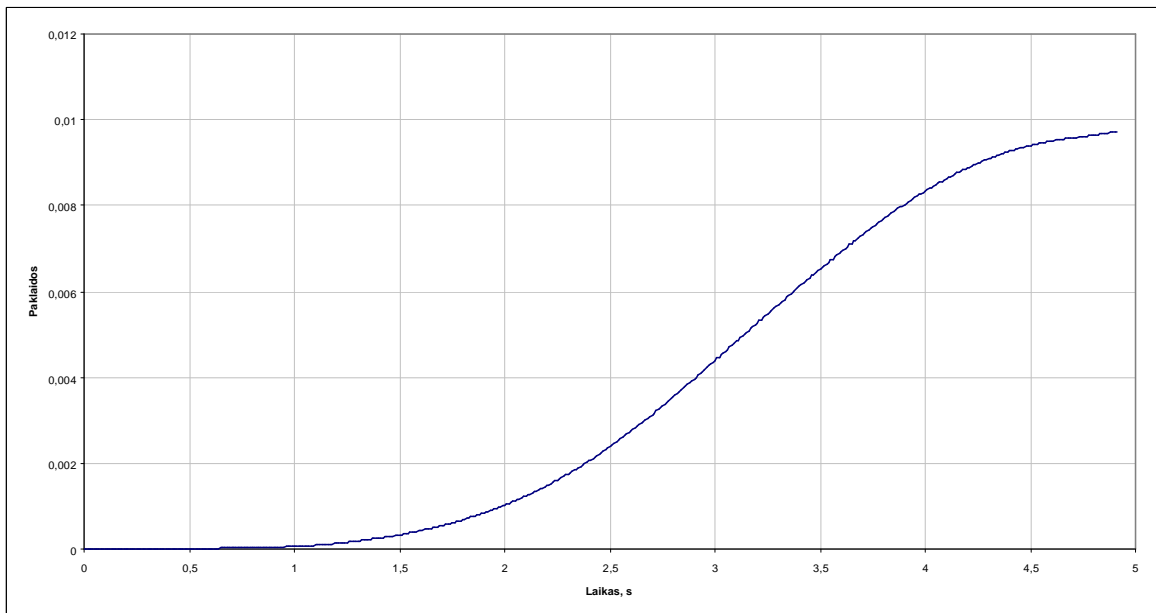
Kaip matyti iš 2.7.5 ir 2.7.6 paveikslų, atrodytų, jog apie deformacijos pradžia galime spręsti nagrinėdami paklaidas mažojo intervaliuko galuose ir analizuodami paklaidos padidėjimą, tačiau toks metodas netinka, nes nors esant deformacijai paklaida padidėja kelis kartus, tačiau tuose intervaliukuose, kur paklaidų grafikas kerta absicės ašį, toks padidėjimas gali siekti kelis šimtus ar net kelis tūkstančius kartų net jei deformacijos ir nėra. Tad čia susiduriame su deformacijos fiksavimo kriterijaus, kuris tiktu bendruoju atveju, problema.

Sudarykime naują f-ją:

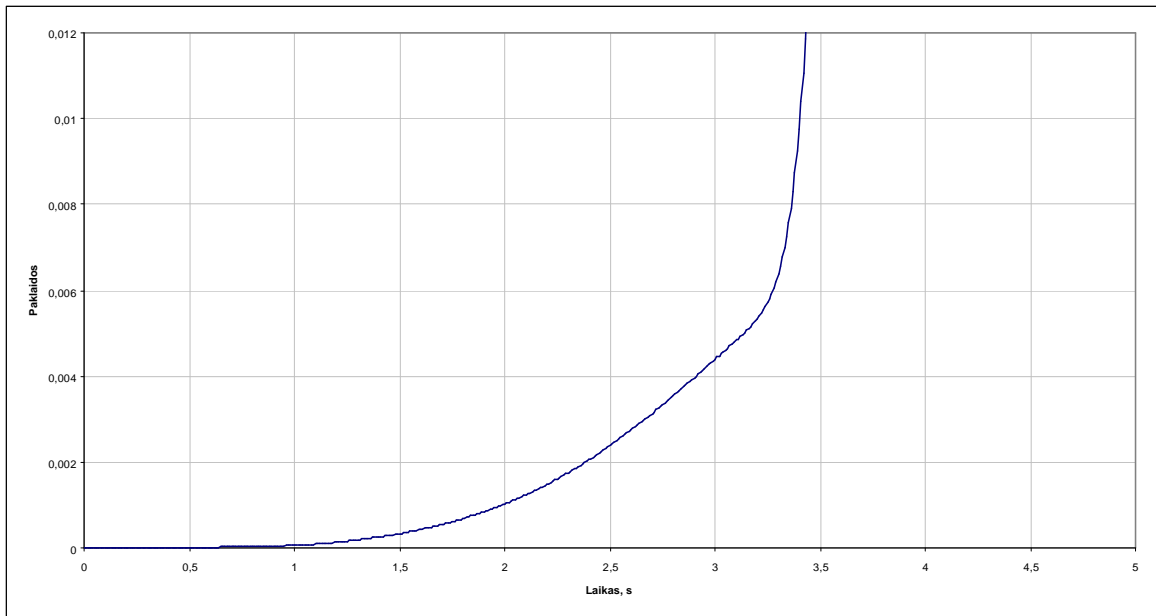
$$f(t_k) = \sum_0^{i=k} |p(t_i)| \quad (2.7.1)$$

čia  $p(t_i)$  - paklaida laiko momentu  $t_i$ .

Šios suminės f-jos grafikas trajektorijai A1+B1 pateiktas 2.7.7 paveiksle.

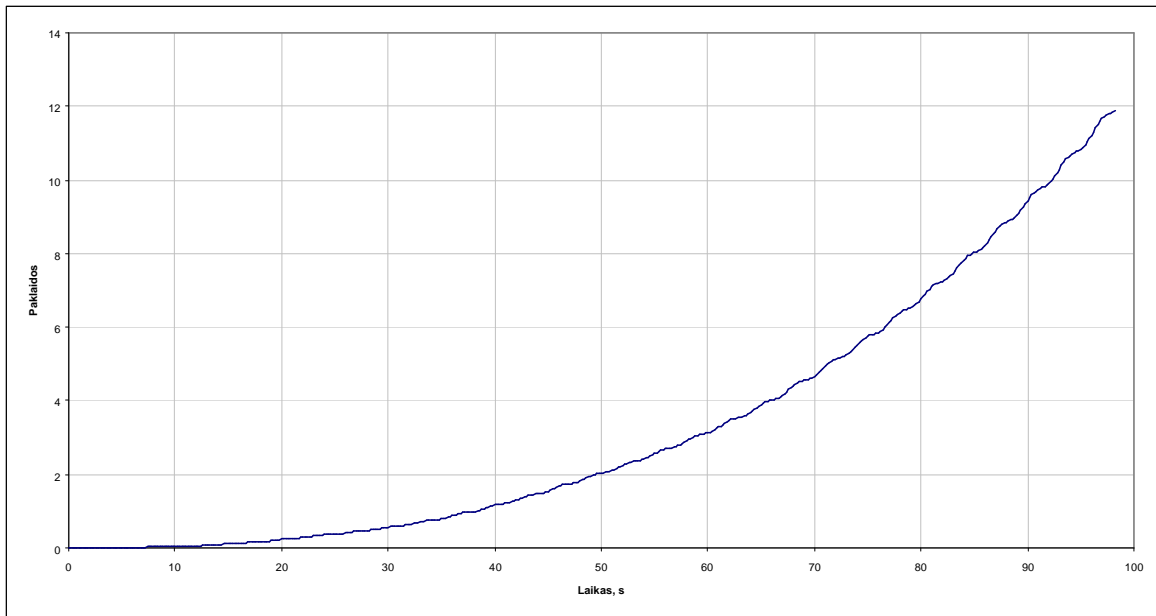


2.7.7 pav.  $f(t_k)$  grafikas trajektorijai A1+B1 iki  $t = 5$  s

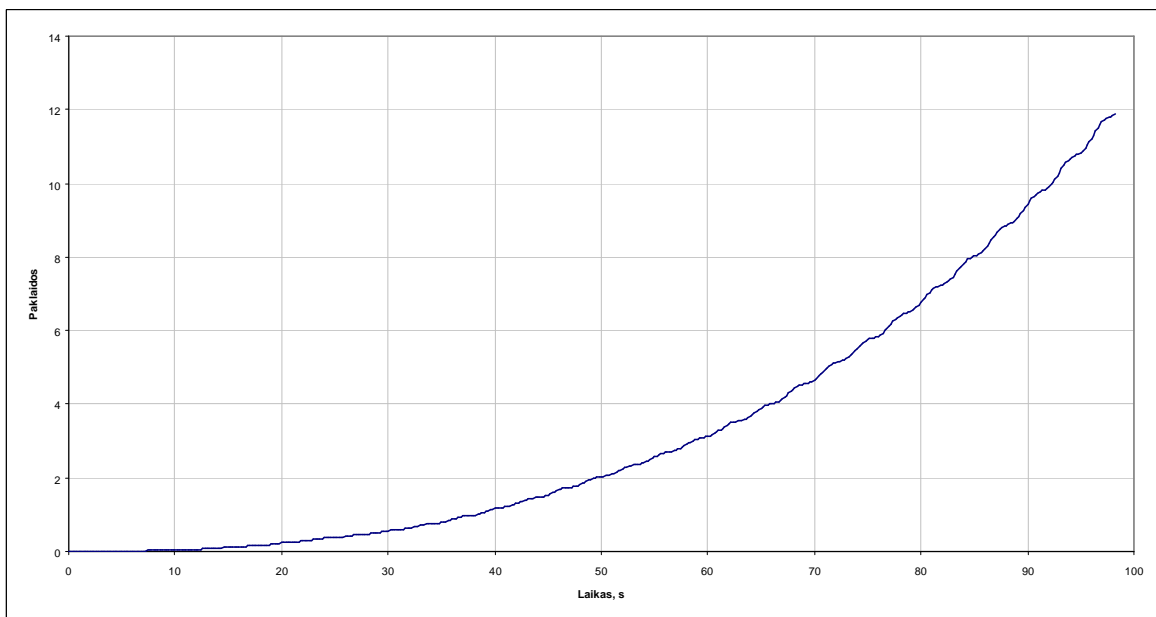


**2.7.8 pav.  $f(t_k)$  grafikas trajektorijai A1+B1 iki  $t = 5$  s deformacijai įvykus laiko momentu  $t = 3$  s**

Atrodytų, jog apie deformacijos pradžią galima būtų spręsti pagal  $f(x)$  pokytį per 0,1 sekundės, tačiau čia problema yra ta, kad nėra tokio universalus apibrėžimo  $f(x)$  pokyčiui, kuris leistų užfiksuoti deformacijas bet kokio ilgio intervale, t.y. jei nagrinėjant trajektorijos A1+B1 suminę funkciją  $f(x)$  iki  $t = 5$  s su deformacija ties  $t = 3$  s, jai (deformacijai) fiksuoti 30%  $f(x)$  prieaugio limitas yra tinkamas, tai nagrinėjant intervalą iki  $t = 100$  s su deformacija ties  $t = 80$  s,  $f(x)$  prieaugis per 0,1s tokio barjero neperžengs ir deformacija nebus fiksuojama. Žemesnio barjero imti irgi negalime, nes tuomet fiksuojamos deformacijos ten, kur jų nėra, dažniausiai nagrinėjamo intervalo pradžioje, kai suminė funkcija  $f(x)$  įgyja sąlyginai mažas reikšmes lyginant su paklaidų vertėmis.



**2.7.9 pav.  $f(t_k)$  grafikas trajektorijai A1+B1 iki  $t = 100$  s**



**2.7.10 pav.  $f(t_k)$  grafikas trajektorijai A1+B1 iki  $t = 100$  s deformacijai įvykus laiko momentu  $t = 80$  s**

Minėtą faktą puikiai pademonstruoja 2.7.9 ir 2.7.10 paveikslai. Kaip matyti,  $f(x)$  grafikas, kai nėra deformacijos, praktiškai nesiskiria nuo grafiko, kai ji yra.

Apibrėžkime dar vieną funkciją, kurios pagalba konstruosime kriterijų:

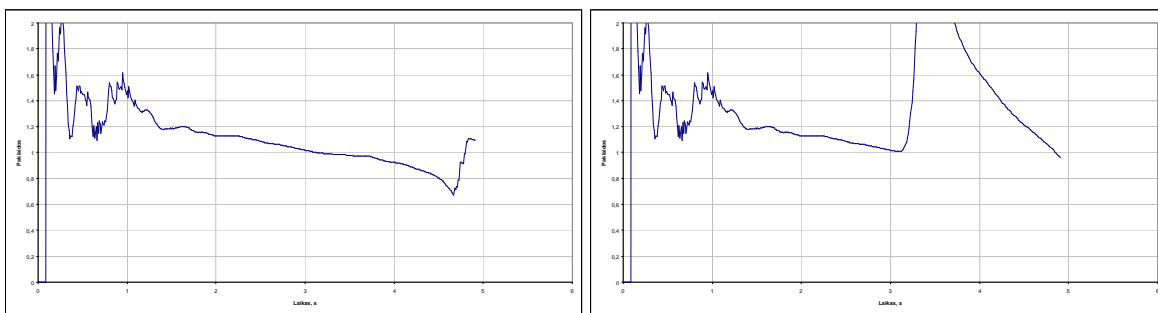
$$d(t_i) = f(t_i) - f(t_{i-9}) \quad (2.7.2)$$

Ši skirtuminė funkcija parodo paklaidų sumos prieaugį per nagrinėjamą interвалиuką, kurio ilgis yra 0,1s. Kaip rodo patirtis, bet kuri funkcija, kuri turi tendencijas didėti su laiku kriterijui netinka, todėl mums reikalinga tokia funkcija, kuri būtų arba pastovi, arba svyruojanti apie kažkokią tai pastovią reikšmę nedidėjančia amplitude. Apibrėžkime santykio f-ją

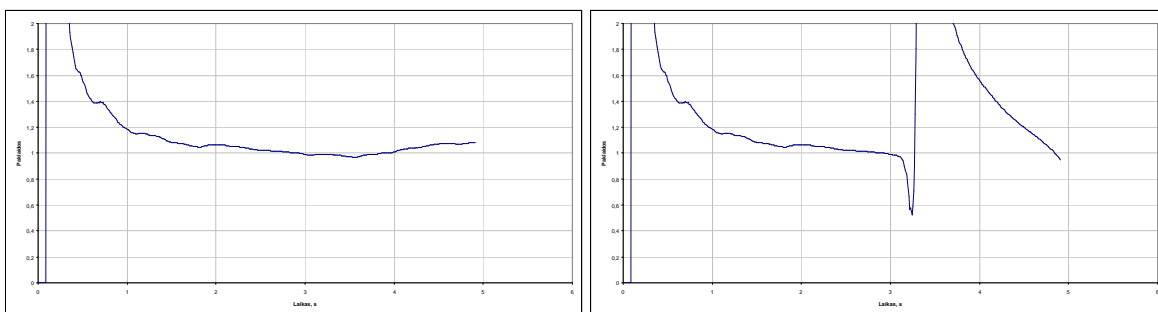
$$s(t_i) = \frac{d(t_i)}{d(t_{i-9})} \quad (2.7.3)$$

kuri ir bus deformacijos fiksavimo kriterijus.

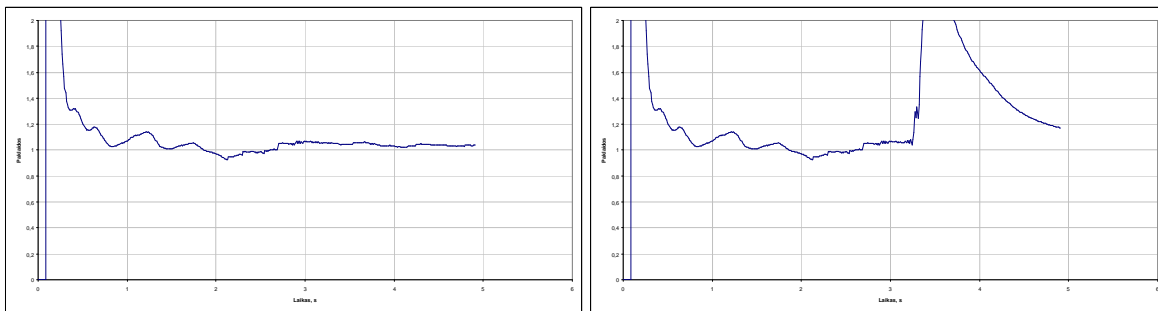
Šios funkcijos grafikas trajektorijoms A1+B1, A2+B2 ir A3+B3 iki laiko  $t = 5$  s pateiktas žemiau esančiuose paveiksluose.



**2.7.11 pav.  $s(t_i)$  grafikas trajektorijai A1+B1 iki  $t = 5$  s be deformacijos ir deformacijai įvykus laiko momentu  $t = 3$  s**

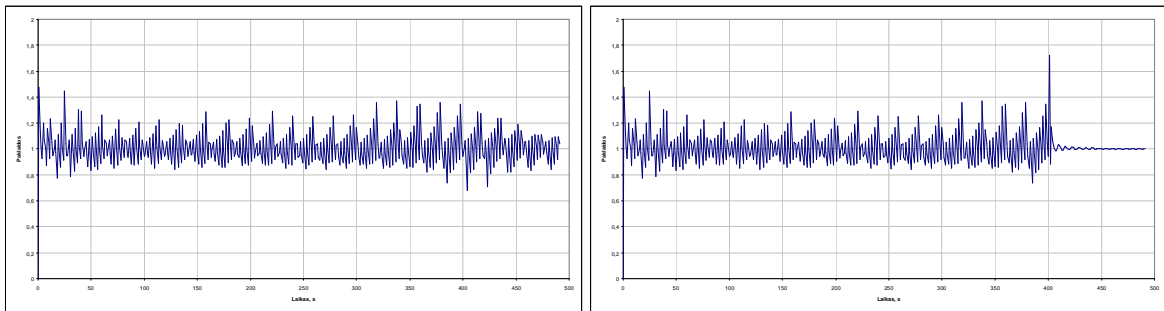


**2.7.12 pav.  $s(t_i)$  grafikas trajektorijai A2+B2 iki  $t = 5$  s be deformacijos ir deformacijai įvykus laiko momentu  $t = 3$  s**

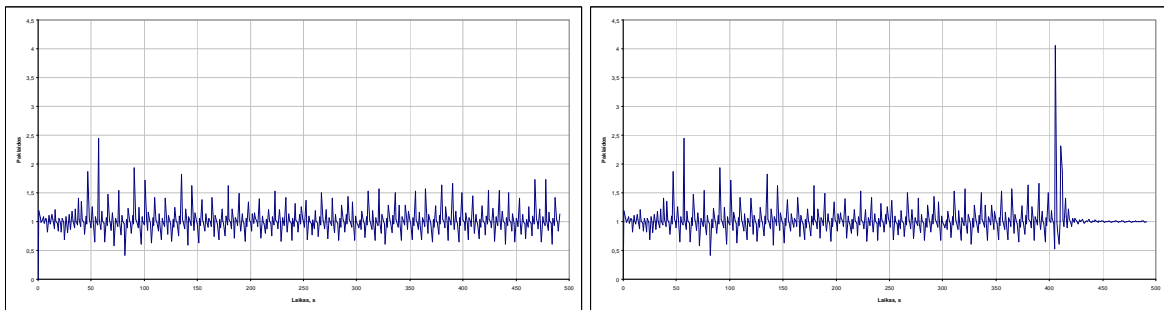


**2.7.13 pav.  $s(t_i)$  grafikas trajektorijai A3+B3 iki  $t = 5$  s be deformacijos ir deformacijai įvykus laiko momentu  $t = 3$  s**

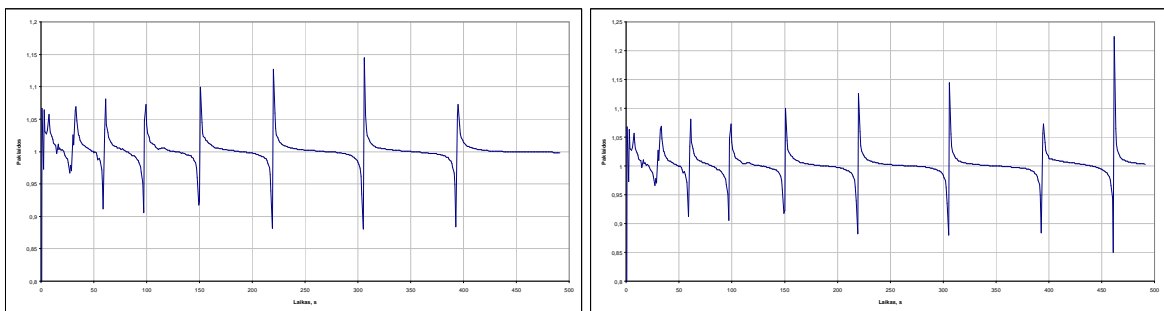
Kaip matyti iš 2.7.11 - 2.7.13 paveikslų, įvykus deformacijai stebime  $s(t_i)$  šuolį, kuris ir bus naudojamas deformacijai fiksuoti. Šiuo atveju nustatomas intervalas apie vieneta, kuriame svyruoja  $s(t_i)$  reikšmės, ir jei  $s(t_i)$  reikšmė „iššoka“ už intervalo ribų – fiksuojama deformacija. 2.7.14 - 2.7.16 paveiksluose pateikti  $s(t_i)$  grafikai toms pačioms trajektorijoms, tik iki laiko momento  $t = 500$  s ir deformacijos pradžia ties  $t = 400$  s.



**2.7.14 pav.  $s(t_i)$  grafikas trajektorijai A1+B1 iki  $t = 500$  s be deformacijos ir deformacijai įvykus laiko momentu  $t = 400$  s**



**2.7.15 pav.  $s(t_i)$  grafikas trajektorijai A2+B2 iki  $t = 500$  s be deformacijos ir deformacijai įvykus laiko momentu  $t = 400$  s**



**2.7.16 pav.  $s(t_i)$  grafikas trajektorijai A2+B2 iki  $t = 500$  s be deformacijos ir deformacijai įvykus laiko momentu  $t = 400$  s**

Kaip matyti iš 2.7.11 - 2.7.13 paveikslų, nors funkcija  $s(t_i)$  yra svyruojanti apie vieneta, tačiau svyravimų amplitudė iki 1-2s yra labai didelė, todėl turi būti atmetama. Čia įvedame dar vieną prielaidą: deformacijų sistemos darbo pradžioje nesitikime. Deformacijų fiksavimo rezultatai visoms trajektorijoms skirtingiems laiko intervalams pateikti žemiau esančioje lentelėje. Čia deformacija įvyksta ties nagrinėjamo intervalo viduriu ir deformuoja atstumą tarp taškų nuo 1 iki 2. Lentelėje taip pat pateikiamos ribos, kurios buvo parinktos funkcijai  $s(t_i)$ .

**2.7.1 lentelė**

**Deformacijos fiksavimo laikai, kai deformacijos pradžia yra per nagrinėjamo intervalo vidurį**

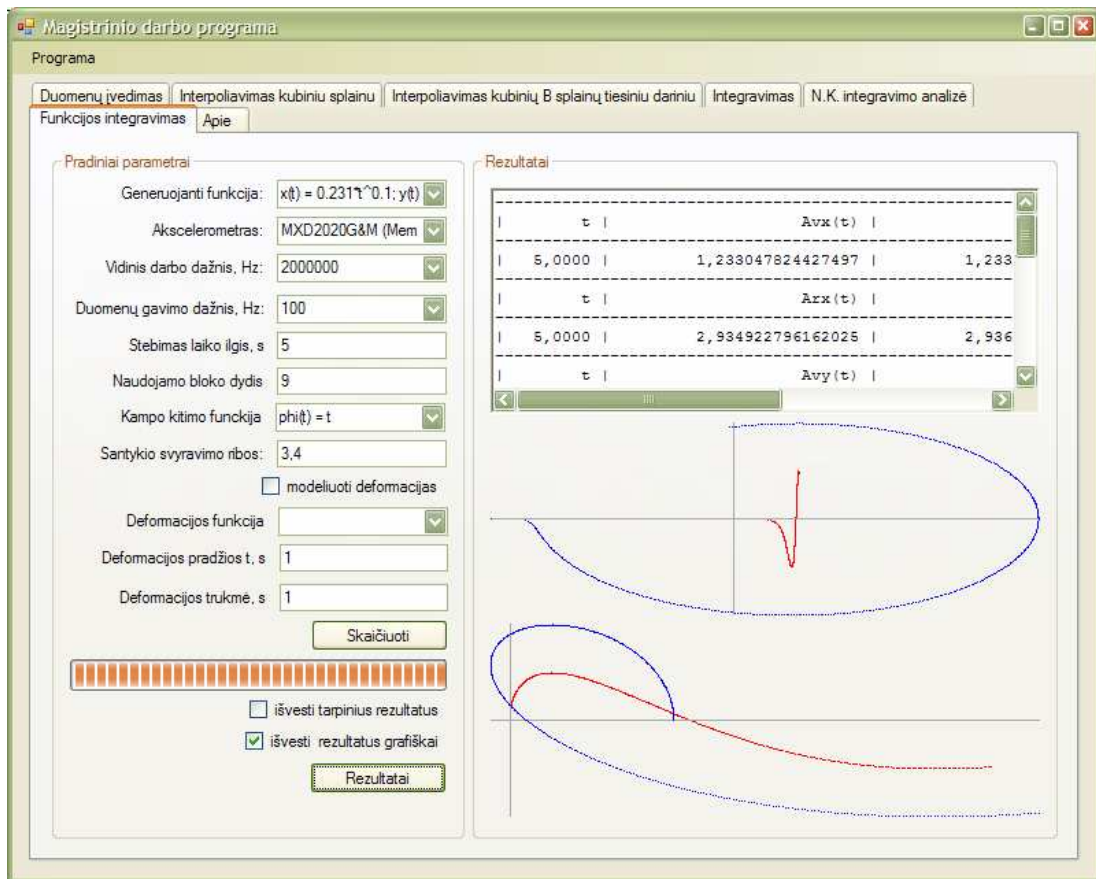
Trajekt.	Nuokrypis nuo 1	Nagrinėjamas intervalas, s								
		50			100			500		
		1	2	3	1	2	3	1	2	3
A1+B1	0,55	25,3800	25,1700	25,1000	50,5000	50,1800	50,1200	252,2000	251,9300	251,7600
A1+B2	2,39	27,2900	25,1600	25,1200	50,4900	50,2900	50,2200	251,1800	251,7000	253,6700
A1+B3	0,21	25,6700	25,3500	25,2300	75,0900	74,7900	74,5900	299,0900	298,8400	298,6600
A2+B1	0,55	25,3800	25,1700	25,1000	50,5000	50,1800	50,1200	252,2000	251,9300	251,7600
A2+B2	2,39	27,2900	25,1600	25,1200	50,4900	50,2900	50,2200	251,1800	251,7000	253,6700
A2+B3	0,21	25,6700	25,3500	25,2300	75,0900	74,7900	74,5900	299,0900	298,8400	298,6600
A3+B1	0,55	25,3800	25,1700	25,1000	50,5000	50,1800	50,1200	252,2000	251,9300	251,7600
A3+B2	2,39	27,2900	25,1600	25,1200	50,4900	50,2900	50,2200	251,1800	251,7000	253,6700
A3+B3	0,21	25,6700	25,3500	25,2300	75,0900	74,7900	74,5900	299,0900	298,8400	298,6600
Atstumas tarp taškų užfiksuotu deformacijos laiko momentu										
A1+B1		1,054872	1,170000	1,316228	1,125000	1,180000	1,346410	2,000000	2,000000	2,000000
A1+B2		2,000000	1,160000	1,346410	1,117649	1,290000	1,469042	2,000000	2,000000	2,000000
A1+B3		1,300763	1,350000	1,479583	2,000000	2,000000	2,000000	2,000000	2,000000	2,000000
A2+B1		1,054872	1,170000	1,316228	1,125000	1,180000	1,346410	2,000000	2,000000	2,000000
A2+B2		2,000000	1,160000	1,346410	1,117649	1,290000	1,469042	2,000000	2,000000	2,000000
A2+B3		1,300763	1,350000	1,479583	2,000000	2,000000	2,000000	2,000000	2,000000	2,000000
A3+B1		1,054872	1,170000	1,316228	1,125000	1,180000	1,346410	2,000000	2,000000	2,000000
A3+B2		2,000000	1,160000	1,346410	1,117649	1,290000	1,469042	2,000000	2,000000	2,000000
A3+B3		1,300763	1,350000	1,479583	2,000000	2,000000	2,000000	2,000000	2,000000	2,000000

### 3 Programinė realizacija ir instrukcija vartotojui

Programa parašyta C# kalba. Jos vykdymui reikalinga Microsoft Windows XP/2000 operacinė sistema su Microsoft .NET 2.0 technologijos palaikymu (papildomas priedas, kurį iš Microsoft galima parsisiųsti ir įdiegti nemokamai).

Modeliavimo ir analizės programos darbo langas pateiktas 3.7.1 paveiksle.





**3.7.1 pav. Programos pagrindinis darbo langas**

Programos puslapių paskirtis tokia:

1. „Duomenų įvedimas“ – generuojamos pasirinktos funkcijos reikšmės remiantis įvestais parametrais.
2. „Interpoliavimas kubiniu splainu“ – puslapis, skirtas skaičiuoti interpoliaciniam polinomui remiantis „Duomenų įvedimas“ puslapyje sugeneruotomis pasirinktos f-jos reikšmėmis.
3. „Interpoliavimas kubinių B splainų tiesiniu dariniu“ – puslapis, skirtas skaičiuoti trečios eilės B splainų tiesiniam dariniui remiantis „Duomenų įvedimas“ puslapyje sugeneruotomis pasirinktos f-jos reikšmėmis.
4. „Integravimas“ – integravimas 1.2 skyriuje išnagrinėtais metodais. Integruojama remiantis 1, 2, 3 puslapių duomenimis ir skaičiavimais.
5. „N.K. integravimo analizė“ – puslapis, skirtas pažiūrėti, kaip keičiasi paklaidos integruojant Niutono ir Koteso formulėmis, kai keičiamas aproksimuojančio polinomo laipsnis.
6. „Funkcijos integravimas“ – pagrindinis analizės ir modeliavimo puslapis, kurio įvedimo laukų paaiškinimai pateikti žemiau.

Puslapio „Funkcijos integravimas“ įvedimo laukų paskirtis:

1. „Generuojanti funkcija“ – pasirenkama taško A pagreičius generuojanti funkcija.
2. „Akselerometras“ – pasirenkamas akselerometro modelis.
3. „Vidinis darbo dažnis, Hz“, „Duomenų gavimo dažnis, Hz“ – akselerometro parametru pasirinkimas.
4. „Stebimas laiko ilgis, s“ – stebimo ir nagrinėjamo intervalo ilgis laiko prasme.
5. „Naudojamo bloko dydis“ – pasirenkamas mažųjų intervaliukų ilgis.
6. „Kampo funkcija“ – pasirenkama taško B judėjimą apie tašką A apibrėžianti funkcija.
7. „Santykio svyravimo ribos“ – įvedama riba funkcijai  $s(t_i)$  (2.6.3), kurią peržengus, fiksuojama deformacija.
8. „Modeliuoti deformacijas“ – įjungiamas deformacijų generavimo mechanizmas.
9. „Deformacijos funkcija“ – deformacijas apibrėžianti funkcija.
10. „Deformacijos pradžios t, s“ – įvedamas deformacijos pradžios laiko momentas.
11. „Deformacijos trukmė, s“ – nurodoma, kiek laiko trunka deformacija.
12. „Išvesti tarpinius rezultatus“ – įjungiamas visų tarpinių duomenų išvedimas rezultatų lange.
13. „Išvesti rezultatus grafiškai“ – įjungiamas scheminis rezultatu išvedimas grafiškai. Išvedami pagreičių funkcijų bei trajektorijų grafikai.

## 4 Diskusija

Pirmiausia verta akcentuoti, kad pasirinktų akselerometrų matavimo tikslumas svyruoja nuo 0,25mg iki 4,0mg, kas, jei  $g$  traktuotume kaip  $10\text{m/s}^2$ , tolygu pagreičio matavimo paklaidoms nuo  $0,0025\text{m/s}^2$  iki  $0,04\text{m/s}^2$ . Turint tokius duomenis, net su itin tiksliu skaičiavimo mechanizmu galimos didelės rezultatų paklaidos.

Duomenų integravimui buvo pasirinktos Niutono ir Koteso kvadratūrinės formulės, tyrinėta paklaidų priklausomybė nuo interpoliacinio polinomo laipsnio  $n$ . 2.4.2 paveiksle stebimas labai įdomus paklaidų elgesys. Pastebėta, kad paklaidos ženkliai didėja, kai polinomo laipsnis yra lyginis.

Pasirinkus darbinius akselerometro darbo parametrus ir atlikus paklaidų priklausomybės nuo interpoliacinio polinomo laipsnio  $n$ , sprendimas dėl  $n$  reikšmės priimtas vertinant ne tik tai, ties kuriuo  $n$  buvo gaunama mažiausia paklaida, bet taip pat buvo žiūrima bendras paklaidų elgesys kuo platesniame laiko intervale. Bet net ir pasirinkus optimaliausią laipsnį, būtų gana drąsu teigti, jog tai visais atvejais yra geriausias variantas, ypač analizuojant pradinius duomenis su tokiomis didelėmis paklaidomis.

Atlikus trajektorijų atkūrimo tyrimą, gauti rezultatai patvirtina modelio tinkamumą, kartu stebime įdomius paklaidų charakterius, diktuojamus periodinių trajektorijas generuojančių funkcijų. Atlikus paklaidų tyrimą po valandos laiko (2.5.1 lentelė) galima išskirti tokių akselerometrų sistemos pritaikymo sritis, t.y. jei jie daugiau mažiau tinkami sekti objektams su vidutine dinamika, tai jei gali visiškai netikti objektas su intensyvia dinamika – vibracijoms, dažnai kintančiam judėjimui (kažkas panašaus į musės skrydžio trajektoriją).

Atliekant stebimas įdomus deformacijų fiksavimo kriterijaus elgesys (2.6 skyrius). Įdomu stebėti, jog tai yra iš viršaus ir apačios aprėžta funkcija (tai grindžiama tik pateiktais grafikais), leidžianti anksčiau ar vėliau užfiksuoti atsiradusią deformaciją, nors pačios paklaidos gali pastoviai didėti. Čia labai įdomu būtų pažiūrėti kriterijaus tinkamumą įvairaus pobūdžio realioms duomenis.

## Išvados

1. Sudarytas problemos matematinis modelis, paruoštas duomenų generavimo mechanizmas pagal pasirinktus nagrinėjimui akselerometrus. Nustatyta, kad turint kelių griežtai susijusių taškų sistemą, bet kuris iš jų gali būti traktuojamas, kaip rotacijos centras, apie kurį kažkokiu dėsnio juda likę taškai.
2. Skaitiniam integravimui pasirinktos devintos eilės Niutono ir Koteso kvadratūrinės formulės, nes šiuo metodu integruojant gaunamos gerokai mažesnės integravimo paklaidos, lyginant su integravimu naudojant kubinius interpoliacinius splainus bei trečios eilės B splainų tiesinius darinius.
3. Suformuota integravimo strategija. Atlikus paklaidų priklausomybės nuo akselerometro darbo parametrų, pasirinktas modelis, matavimus atliekantis 100Hz dažniu ir dirbantis 2MHz vidinio laikrodžio dažniu. Šis modelis, pagal gamintojo specifikaciją yra jautriausias iš analizei pasirinktų, tą patvirtino ir tyrimas su modeliuotais duomenimis.
4. Nustatyta, kad modelis atkuria pradines trajektorijas, tačiau procesui vykstant ilgą laiką, vibracijas primenančių trajektorijų atkūrimas nukrypsta nuo pradinės. Tai rodo prietaiso netinkamumą tokio tipo judesiui analizuoti.
5. Sudarytas deformacijos fiksavimo kriterijus. Atliktas tyrimas parodė, kad nagrinėjant laiko intervalą iki 500 sekundžių, deformacijos fiksavimas gali vėluoti iki 2 sekundžių, kai nagrinėjamas panašus į slenkamą judesys, ir net iki 50 sekundžių – kai judesys yra vibracinio pobūdžio.

## **Rekomendacijos**

Nors darbe išanalizuota daug duomenų, tačiau visi jie yra modeliavimo rezultatas, todėl tolimesniems tyrimams būtini realūs duomenys: pavyzdžiui surinkti važinėjant automobiliu po miestą.

Buvo tiriama dviejų taškų sistema plokštumoje. Tolimesnis tyrimas galėtų būti atliekamas trimatėje erdvėje stebint gerokai daugiau taškų ir analizuojant jų duomenis ir tarpusavio ryšius.

Darbe rinktasi iš trijų skaitinio integravimo metodų, tačiau čia galima atlikti platesnę metodų analizę.

Sėkmingai gali būti tiriamas ir sistemos ekonomiškumas, t.y. skirtingų akselerometrų parinkimas pagal reikalaujamą tikslumą ar taikymo srities pobūdį.

## Literatūra

1. National Instruments – Developer Zone / Accelerometer Principles. <http://zone.ni.com/devzone/cda/ph/p/id/12> [žiūrėta 2007 05 10].
2. National Instrument – Developer Zone / Types of Accelerometers. <http://zone.ni.com/devzone/cda/ph/p/id/284> [žiūrėta 2007 05 10].
3. Aszkler, C. The Principles of Acceleration, Shock And Vibration Sensors. Sensors. 5/1/2005. <http://www.sensorsmag.com/sensors/article/articleDetail.jsp?id=184885&searchString=accelerometer> [žiūrėta 2007 05 10]
4. Plūkas, K. Skaitiniai metodai ir algoritmai. Kaunas 2001.
5. <http://www.memsic.com/memsic/products/gdxxi.asp?pq=111> [žiūrėta 2007 05 11].
6. Benevičius, V.; Listopadskis, N. Akselerometrų taikymo matematinis modelis, Lietuvos matematikos rinkinys : Lietuvos matematikų draugijos XLVII konferencijos mokslo darbai / Matematikos ir informatikos institutas, Lietuvos matematikų draugija, Vilniaus universitetas. ISSN 0132-2818. Vilnius : Matematikos ir informatikos institutas, 2006, T. 46, spec. nr, p. 350-355.
7. Benevičius, V.; Listopadskis, N. Akselerometrų taikymo matematinis modelis, Matematika ir matematinis modeliavimas / Kauno technologijos universitetas. ISSN 1822-2757. Kaunas : Technologija. T. 2 (2006), p. 131-135.
8. Benevičius, V.; Listopadskis, N. Matematinio modelio, judesio trajektorijai nustatyti remiantis akselerometrų duomenimis, sudarymas ir tyrimas, Taikomoji matematika : VI studentų konferencijos pranešimų medžiaga / Kauno technologijos universitetas. ISBN 9955-25-044-5. Kaunas : Technologija, 2006, p. 10-11.

## 1 PRIEDAS. PROGRAMOS TEKSTAS

Čia pateikiamos pagrindinės klasės ir kodas.

NiutonoKotesoFormules.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace SkMtd
{
    public class NiutonoKotesoFormules
    {
        private Vektorius y; //y tinklelis
        private Vektorius w; //koeficientai w
        private int n; //kvadraturiniu formuliu eile
        private Decimal h; //integravimo zingsnis arba mastelis

        public NiutonoKotesoFormules()
        {
        }
        public void NustatytiTinkleli(Vektorius y)
        {
            this.y = y;
        }
        private Decimal Mk(Decimal k)
        {
            return (1m / (k + 1m));
        }
        public Decimal Reiksme()
        {
            Decimal rez = 0;
            for (int ii = 0; ii <= this.n; ii++)
            {
                rez += this.w[ii] * this.y[ii];
            }
            return (rez);
        }
        public Decimal Reiksme(Decimal dt)
        {
            Decimal rez = 0;
            for (int ii = 0; ii <= this.n; ii++)
            {
                rez += this.w[ii] * this.y[ii];
            }
            return (dt / this.h * rez);
        }
        //*****
        public void Skaiciuoti(int n) //apskaiciuoja wi pagal tinkleli tarp 0 ir 1
        {
            this.n = n;
            this.h = 1m / (Decimal)this.n;
            Vektorius x = new Vektorius(this.n + 1);
            for (int ii = 0; ii <= this.n; ii++) x[ii] = this.h * (Decimal)ii;
            Matrica W = new Matrica(this.n + 1, this.n + 1); //kuriam sistema matrica
            Vektorius xk = new Vektorius(this.n + 1); //vektorius xk laipsniams saugoti
            for (int ii = 0; ii <= this.n; ii++) xk[ii] = 1; //visur vienetai, t.y. xk^0
            Vektorius B = new Vektorius(this.n + 1); //laisvasis stulpelis
            for (int ll = 0; ll <= this.n; ll++) //formuojam lygtis
            {
                for (int ii = 0; ii <= this.n; ii++) //koeficientu irasymas
                {
                    W[ll][ii] = xk[ii];
                    xk[ii] = xk[ii] * x[ii]; //perskaiciuojamas laipsnis
                }
            }
        }
    }
}
```

```

    }
    B[11] = this.Mk(11); //irasomas 11 momentas
}
GausoMetodas gm = new GausoMetodas();
gm.IrasytiC(W);
gm.IrasytiB(B);
gm.Sprest();
this.w = gm.Sprendinys; //randami koeficientai
}
}
}

```

### Integravimas.cs

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Threading;
using System.Runtime.Remoting.Messaging;
using System.Windows.Forms;
using System.ComponentModel;
using SkMtd;

namespace magistras
{
    public class Integravimas
    {
        ProgressBar prg; //progress baras, kuris rodo proceso eiga
        ComboBox CCR; //CClockRates Combo
        ComboBox SR; //SampleRates Combo
        Button RezBtn; //rezultatu mygtukas
        Vektorius t;
        Vektorius[][] d; //duomenu lentele
        int i_f; //funkcijos combo boxe pasirinkimas
        int i_phi; //kampo kitimo funkcija
        Decimal dt; //zingsnis dt
        int N; //zingsniu kiekis
        public Akselerometras acc; //akscelerometro klase
        bool IsvestiTarpinius;
        int BlokoDydis; //po kiek intervalu narginejama
        int i_def; //deformacijos funkcija
        Decimal defT0; //deformacijos prazia
        Decimal defT; //deformacijos trukme
        Decimal defK; //deformacijos fiksavimo konstanta
        bool UseDef; //ar deformacija naudojamas
        Vektorius R; //spinduliu masyvas
        Vektorius RPakl; //spinduliu paklaidu masyvas
        Vektorius MRR; //vidurkiu vidurkis
        Vektorius MR; //paklaidu vidurkiai blokais
        Vektorius SumR; //paklaidu suma
        Vektorius VSumR; //paklaidu didejimo greitis
        Vektorius VSant; //santykis tarp greiciu
        Decimal R0;
        int di;

        public Integravimas()
        {
            this.R0 = 1m;
        }
        public void NustatytiRysiSuForma(ProgressBar prg, ComboBox CCR, ComboBox SR, Button
rez)
        {
            this.prg = prg;
            this.CCR = CCR;
            this.SR = SR;

```



```

        this.RezBtn = rez;
    }
    public void NustatytiParametrus(int fi, int phii, int CCR, int SR, int i_sk, Decimal
defk) //kuri funkcija
    {
        if (acc == null) throw (new Exception("Nepasirinktas akscelerometras"));
        this.i_f = fi;
        this.acc.CCR = this.acc.CClockRates[CCR];
        this.acc.SR = this.acc.SampleRates[SR];
        this.BlokoDydis = i_sk;
        this.i_phi = phii;
        this.UseDef = false;
        this.defk = defk;
    }
    public void NustatytiDeformacijosParametrus(int fi, Decimal deft0, Decimal deft)
    {
        this.i_def = fi;
        this.deft0 = deft0;
        this.deft = deft;
        this.UseDef = true;
    }
    delegate bool IntegravimasDelegate();
    public bool Integruoti() //metodas, kuris integruoja tinkelli du kartus
    {
        //*****
        //duomeni paruosimas
        //*****
        this.dt = 1m / this.acc.SR; //laiko pokytis duomeni paemimui
        if (this.d == null)
        {
            this.d = new Vektorius[4][];
            for (int ii = 0; ii < 4; ii++) this.d[ii] = new Vektorius[3];
        }
        for (int ii = 0; ii < 4; ii++)
        {
            for (int jj = 0; jj < 3; jj++)
            {
                this.d[ii][jj] = new Vektorius(this.N + 1);
            }
        }

        this.t = new Vektorius(this.N + 1);
        this.R = new Vektorius(this.N + 1);
        this.RPakl = new Vektorius(this.N + 1);
        this.MR = new Vektorius(this.N + 1);
        this.MRR = new Vektorius(this.N + 1);
        this.SumR = new Vektorius(this.N + 1);
        this.VSumR = new Vektorius(this.N + 1);
        this.VSant = new Vektorius(this.N + 1);
        int defi = 0;
        for (int ii = 0; ii <= this.N; ii++) //sudaroma reiksmiu lentele
        {
            this.t[ii] = (Decimal)ii * this.dt;
            this.d[0][0][ii] = this.acc.Reiksme(Funkcijos.Aax(this.i_f, this.t[ii]));
            this.d[1][0][ii] = this.acc.Reiksme(Funkcijos.Aay(this.i_f, this.t[ii]));
            this.R[ii] = this.R0; //pradinis spindulys
            if (this.UseDef) //jei naudojama deformacija
            {
                if ((this.t[ii] >= this.deft0) && (this.t[ii] <= this.deft0 + this.deft))
//deformacija vyksta
                {
                    this.R[ii] = this.R0 + Funkcijos.R(this.i_def, this.t[ii] -
this.deft0);
                    defi = ii;
                }
                if (this.t[ii] > this.deft0 + this.deft) this.R[ii] = this.R[defi];
            }
        }
    }

```

```

    }
    this.d[2][0][ii] = this.d[0][0][ii] + this.acc.Reiksme(this.R[ii] *
Funkcijos.Bax(this.i_phi, this.t[ii]));
    this.d[3][0][ii] = this.d[1][0][ii] + this.acc.Reiksme(this.R[ii] *
Funkcijos.Bay(this.i_phi, this.t[ii]));
    }
    this.prg_PerformStep(); //parodom, kad sugeneravom duomenis
    /*******
    //pirmasis integravimo ciklas
    /*******
    int i_sk = this.BlokoDydis; //po kiek intervalu skaidom
    int npr;
    Vektorius[][] ab = new Vektorius[4][];
    for (int ii = 0; ii < 4; ii++)
    {
        ab[ii] = new Vektorius[2];
        ab[ii][0] = new Vektorius(i_sk + 1);
        ab[ii][1] = new Vektorius(i_sk);
    }
    NiutonoKotesoFormules nk1 = new NiutonoKotesoFormules(); //integravimo klase
    NiutonoKotesoFormules nk2 = new NiutonoKotesoFormules();
    nk1.Skaiciuoti(i_sk);
    nk2.Skaiciuoti(i_sk - 1);
    Decimal int1, int2;//integralu reiksmes
    Decimal pakl;
    int count;
    this.di = this.N + 2;
    int nsk = (int)this.acc.SR; //nuo kur pradeti skaiciuoti //pirmos sekundes
nestebime
    for (int nn = 0; nn < 2; nn++) //du integravimo zingsniai
    {
        npr = 1;
        while (npr <= this.N) //skaidymas i intervalus
        {
            if (npr < i_sk) //pirmi 12 tasku
            {
                for (int ii = 0; ii <= i_sk; ii++)
                {
                    ab[0][0][ii] = this.d[0][nn][ii + npr - 1]; //paimami mazieji
tinkleliai
                    ab[1][0][ii] = this.d[1][nn][ii + npr - 1];
                    ab[2][0][ii] = this.d[2][nn][ii + npr - 1];
                    ab[3][0][ii] = this.d[3][nn][ii + npr - 1];
                    if (ii < i_sk)
                    {
                        ab[0][1][ii] = this.d[0][nn][ii + npr];
                        ab[1][1][ii] = this.d[1][nn][ii + npr];
                        ab[2][1][ii] = this.d[2][nn][ii + npr];
                        ab[3][1][ii] = this.d[3][nn][ii + npr];
                    }
                }
            }
            nk1.NustatytiTinkleli(ab[0][0]);
            int1 = nk1.Reiksme(this.dt); //integralas per 12 intervalu
            nk2.NustatytiTinkleli(ab[0][1]);
            int2 = nk2.Reiksme(this.dt); //integralas per 11 intervalu
            this.d[0][nn + 1][npr] = this.d[0][nn + 1][npr - 1] + int1 - int2;
//integralo reiksme skaiciuojame taske
            nk1.NustatytiTinkleli(ab[1][0]);
            int1 = nk1.Reiksme(this.dt); //integralas per 12 intervalu
            nk2.NustatytiTinkleli(ab[1][1]);
            int2 = nk2.Reiksme(this.dt); //integralas per 11 intervalu
            this.d[1][nn + 1][npr] = this.d[1][nn + 1][npr - 1] + int1 - int2;
//integralo reiksme skaiciuojame taske

            nk1.NustatytiTinkleli(ab[2][0]);
            int1 = nk1.Reiksme(this.dt); //integralas per 12 intervalu

```

```

        nk2.NustatytiTinkleli(ab[2][1]);
        int2 = nk2.Reiksme(this.dt); //integralas per 11 intervalu
        this.d[2][nn + 1][npr] = this.d[2][nn + 1][npr - 1] + int1 - int2;
//integralo reiksme skaiciuojame taske
        nk1.NustatytiTinkleli(ab[3][0]);
        int1 = nk1.Reiksme(this.dt); //integralas per 12 intervalu
        nk2.NustatytiTinkleli(ab[3][1]);
        int2 = nk2.Reiksme(this.dt); //integralas per 11 intervalu
        this.d[3][nn + 1][npr] = this.d[3][nn + 1][npr - 1] + int1 - int2;
//integralo reiksme skaiciuojame taske
    }
    else //visi kiti tarpiniai
    {
        for (int ii = 0; ii <= i_sk; ii++)
        {
            ab[0][0][ii] = this.d[0][nn][ii + npr - i_sk]; //paimami mazieji
tinkleliai
            ab[1][0][ii] = this.d[1][nn][ii + npr - i_sk];
            ab[2][0][ii] = this.d[2][nn][ii + npr - i_sk];
            ab[3][0][ii] = this.d[3][nn][ii + npr - i_sk];
        }
        nk1.NustatytiTinkleli(ab[0][0]);
        if (npr == i_sk) this.d[0][nn + 1][npr] = nk1.Reiksme(this.dt);
        else this.d[0][nn + 1][npr] = this.d[0][nn + 1][npr - i_sk] +
nk1.Reiksme(dt);
        nk1.NustatytiTinkleli(ab[1][0]);
        if (npr == i_sk) this.d[1][nn + 1][npr] = nk1.Reiksme(this.dt);
        else this.d[1][nn + 1][npr] = this.d[1][nn + 1][npr - i_sk] +
nk1.Reiksme(dt);
        nk1.NustatytiTinkleli(ab[2][0]);
        if (npr == i_sk) this.d[2][nn + 1][npr] = nk1.Reiksme(this.dt);
        else this.d[2][nn + 1][npr] = this.d[2][nn + 1][npr - i_sk] +
nk1.Reiksme(dt);
        nk1.NustatytiTinkleli(ab[3][0]);
        if (npr == i_sk) this.d[3][nn + 1][npr] = nk1.Reiksme(this.dt);
        else this.d[3][nn + 1][npr] = this.d[3][nn + 1][npr - i_sk] +
nk1.Reiksme(dt);
        //greicio pataisa pagal nulini greiti
        if (nn == 0)
        {
            this.d[0][1][npr - i_sk] += Funkcijos.Avx(this.i_f, 0);
            this.d[1][1][npr - i_sk] += Funkcijos.Avy(this.i_f, 0);
            this.d[2][1][npr - i_sk] += Funkcijos.Avx(this.i_f, 0) +
Funkcijos.Bvx(this.i_phi, 0);
            this.d[3][1][npr - i_sk] += Funkcijos.Avy(this.i_f, 0) +
Funkcijos.Bvy(this.i_phi, 0);
        }
        else //arba padeciu korekcija
        {
            this.d[0][2][npr - i_sk] += Funkcijos.Arx(this.i_f, 0);
            this.d[1][2][npr - i_sk] += Funkcijos.Ary(this.i_f, 0);
            this.d[2][2][npr - i_sk] += Funkcijos.Arx(this.i_f, 0) +
Funkcijos.Brx(this.i_phi, 0);
            this.d[3][2][npr - i_sk] += Funkcijos.Ary(this.i_f, 0) +
Funkcijos.Bry(this.i_phi, 0);
            pakl = Mat.Pow(this.d[0][2][npr - i_sk] - this.d[2][2][npr -
i_sk], 2m);
            pakl += Mat.Pow(this.d[1][2][npr - i_sk] - this.d[3][2][npr -
i_sk], 2m);
            pakl = Mat.Sqrt(pakl); //atstumas tarp tasku
            this.RPakl[npr - i_sk] = pakl - this.R0; //absoliutine paklaida
            pakl = 0;
            count = 0;
            for (int ii = 0; ii <= i_sk; ii++)
            {

```

```

        if (npr - i_sk - ii >= 0)
        {
            pakl += this.RPakl[npr - i_sk - ii];
            count++;
        }
    }
    this.MR[npr - i_sk] = pakl / (Decimal)count;
    pakl = 0;
    count = 0;
    for (int ii = 0; ii <= i_sk; ii++)
    {
        if (npr - i_sk - ii >= 0)
        {
            pakl += this.MR[npr - i_sk - ii];
            count++;
        }
    }
    this.MRR[npr - i_sk] = pakl / (Decimal)count;
    if (npr - i_sk > 0)
    {
        this.SumR[npr - i_sk] = this.SumR[npr - i_sk - 1] +
Mat.Abs(this.RPakl[npr - i_sk]);
        //this.VSumR[npr - i_sk] = this.SumR[npr - i_sk] / this.t[npr
- i_sk];
        if (npr - i_sk >= i_sk) this.VSumR[npr - i_sk] = this.SumR[npr
- i_sk] - this.SumR[npr - i_sk - i_sk];
        else this.VSumR[npr - i_sk] = this.SumR[npr - i_sk];
    }
    else this.SumR[npr - i_sk] = Mat.Abs(this.RPakl[npr - i_sk]);
    if (npr - i_sk > i_sk) this.VSant[npr - i_sk] = this.VSumR[npr -
i_sk] / this.VSumR[npr - 2 * i_sk];
    if (npr - i_sk > i_sk)
    {
        if (this.di > this.N + 1)
        {
            //if (Mat.Abs(this.MRR[npr - i_sk]) > Mat.Abs(this.MRR[npr
- i_sk - i_sk + 1]) * this.defk)
            // this.di = npr - i_sk;
            if (this.t[npr - i_sk] > 2m)
                if ((this.VSant[npr - i_sk] > 1 + this.defk) ||
(this.VSant[npr - i_sk] < 1 - this.defk))
                    this.di = npr - i_sk;
        }
    }
    }
    }
    npr++;
}
if (nn == 0)
{
    for (int ii = this.t.Dimensija - i_sk; ii < this.t.Dimensija; ii++)
    {
        this.d[0][1][ii] += Funkcijos.Avx(this.i_f, 0);
        this.d[1][1][ii] += Funkcijos.Avy(this.i_f, 0);
        this.d[2][1][ii] += Funkcijos.Avx(this.i_f, 0) +
Funkcijos.Bvx(this.i_phi, 0);
        this.d[3][1][ii] += Funkcijos.Avy(this.i_f, 0) +
Funkcijos.Bvy(this.i_phi, 0);
    }
}
else
{
    for (int ii = this.t.Dimensija - i_sk; ii < this.t.Dimensija; ii++)
    {
        this.d[0][2][ii] += Funkcijos.Arx(this.i_f, 0);
    }
}

```

```

        this.d[1][2][ii] += Funkcijos.Ary(this.i_f, 0);
        this.d[2][2][ii] += Funkcijos.Arxx(this.i_f, 0) +
Funkcijos.Brx(this.i_phi, 0);
        this.d[3][2][ii] += Funkcijos.Ary(this.i_f, 0) +
Funkcijos.Bry(this.i_phi, 0);
        pakl = Mat.Pow(this.d[0][2][ii] - this.d[2][2][ii], 2m);
        pakl += Mat.Pow(this.d[1][2][ii] - this.d[3][2][ii], 2m);
        pakl = Mat.Sqrt(pakl); //atstumas tarp tasku
        this.RPakl[ii] = pakl - this.R0; //absoliutine paklaida
    }
    }
    this.prg_PerformStep(); //parodom, kad atlikom zingsni
}

return (true);
}
public void PradetiIntegravima(Decimal t, bool tarp) //pradedamas integravimo procesas
{
    this.IsvestiTarpinius = tarp;
    this.dt = 1m / this.acc.SR; //nustatomas dt
    this.N = Decimal.ToInt32(this.acc.SR * t); //nustatom N
    IntegravimasDelegate id = new IntegravimasDelegate(this.Integruoti);
    AsyncCallback cb = new AsyncCallback(this.IntegravimoRezultatai);
    Object state = new Object();
    IAsyncResult ar = id.BeginInvoke(cb, state);
}
public void IntegravimoRezultatai(IAsyncResult ar)
{
    IntegravimasDelegate id = (IntegravimasDelegate)((AsyncResult)ar).AsyncDelegate;
    // Obtain the result.
    id.EndInvoke(ar);
    this.rez_Enabled(true);
}
public Vektorius GetD(int ii, int jj)
{
    if (ii == -1) return (this.t);
    else if (ii == -2) return (this.RPakl);
    else if (ii == -3) return (this.MR);
    else if (ii == -4) return (this.MRR);
    else if (ii == -5) return (this.SumR);
    else if (ii == -6) return (this.VSumR);
    else if (ii == -7) return (this.VSant);
    else if (ii == -8) return (this.R);
    else return (this.d[ii][jj]);
}
public int Getdi()
{
    return (this.di);
}
delegate void PerformStepCallback();
void prg_PerformStep()
{
    if (this.prg.InvokeRequired)
    {
        PerformStepCallback d = new PerformStepCallback(this.prg_PerformStep);
        this.prg.Invoke(d);
    }
    else
    {
        this.prg.PerformStep();
    }
}
delegate void EnableCallback(bool state);
void rez_Enabled(bool state)
{
    if (this.RezBtn.InvokeRequired)

```

```

    {
        EnableCallback d = new EnableCallback(this.rez_Enabled);
        this.RezBtn.Invoke(d, new object[] { state });
    }
    else
    {
        this.RezBtn.Enabled = true;
    }
}
internal void NustatytiAcc(int p)
{
    switch (p)
    {
        case 0:
            this.acc = new MXD6125EF();
            break;
        case 1:
            this.acc = new MXD2020GM();
            break;
        default:
            break;
    }
    Vektorius ccr = this.acc.CClockRates;
    Vektorius sr = this.acc.SampleRates;
    this.CCR.Items.Clear();
    for (int ii = 0; ii < ccr.Dimensija; ii++) this.CCR.Items.Add(ccr[ii]);
    this.SR.Items.Clear();
    for (int ii = 0; ii < sr.Dimensija; ii++) this.SR.Items.Add(sr[ii]);
}
}
}

```