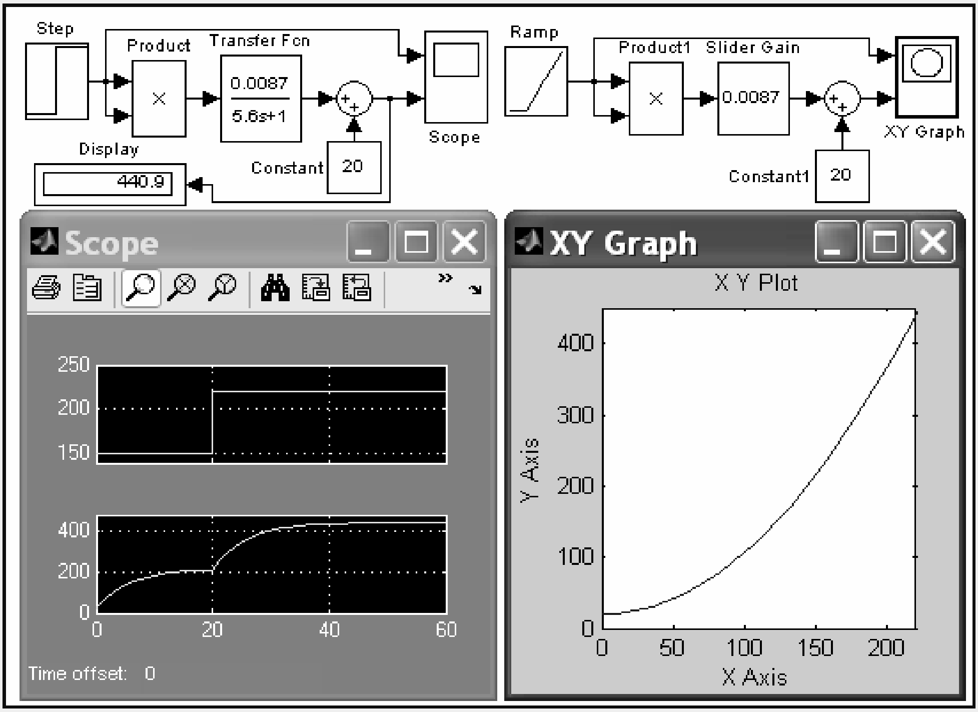
Andris Šnīders

AUTOMĀTISKO SISTĒMU MODELĒŠANA



Jelgava 2008

Latvijas Lauksaimniecības universitāte
Tehniskā fakultāte
Lauksaimniecības enerģētikas institūts

Andris Šnīders

AUTOMĀTISKO SISTĒMU MODELĒŠANA

Mācību līdzeklis lauksaimniecības enerģētikas un informācijas tehnoloģiju specialitātes maģistrantiem

Jelgava 2008

Mācību līdzeklis sagatavots un izdots ESF projekta „Inženierzinātņu studiju satura modernizācija Latvijas Lauksaimniecības universitātē" ietvaros, projektu līdzfinansē Eiropas Savienība.

Šnīders A. Automātisko sistēmu modelēšana: Mācību līdzeklis. – Jelgava: LLU, 2008. – 136 lpp.

Mācību līdzeklis sagatavots atbilstoši LLU lauksaimniecības enerģētikas un informācijas tehnoloģiju specialitāšu maģistra studiju priekšmetu „Automātisko sistēmu modelēšana" un „Ražošanas datorvadības sistēmas 1. daļa. Tehnoloģisko procesu vadības sistēmas" programmām. To varēs izmantot kā palīglīdzekli arī citu LLU inženiertehnisko specialitāšu maģistranti un doktoranti, lai apgūtu tehnoloģisko iekārtu un sistēmu dinamisko procesu imitāciju modelēšanu Windows vidē, izmantojot uz algoritmiskām blokshēmām balstīto programmu MATLAB „Simulink".

Recenzenti:

Rīgas Tehniskās universitātes profesors Dr.habil.sc.ing. Jānis Greivulis.

LEEA valdes loceklis, energofirmas „Jauda" prezidents Jānis Šimins.

ISBN 978-9984-784-63-2

SATURS

IEVADS

Inženierstudiju virsuzdevums ir sagatavot projektu un inovāciju spējīgus inženierus, kuri spēj pilnveidot esošās iekārtas un izstrādāt principiāli jaunas tehnoloģijas. Tā realizācijai nepieciešams: 1 - attīstīt abstrakto analītiski - sintētisko domāšanu, spēju risināt nestandarta problēmuzdevumus specialitātē; 2 - apgūt un pārzināt modernās tehnoloģijas ražošanā, tehnoloģisko procesu vadībā un informācijas apstrādē.

Mūsdienu ražošanas objektīva nepieciešamība ir **tehnoloģisko procesu automatizācija**, kas veicina darba ražīguma celšanu, produkcijas kvalitātes uzlabošanu, izejvielu, materiālu un enerģijas patēriņa samazināšanu, apkalpojošā personāla darba apstākļu uzlabošanu.

Modernā datortehnika un tās aprīkojums dod iespēju realizēt **pilnīgu ražošanas procesa automatizāciju**, kurā visas kontroles un vadības operācijas izpilda automātiskas iekārtas, ieskaitot operatīvo vadību, uzskaiti un plānošanu.

Pilnīgas automatizācijas būtiski svarīga sastāvdaļa ir **tehnoloģisko iekārtu un procesu automātiska vadība**, kuru realizē ar **automātiskās vadības sistēmu (AVS)** palīdzību. Šādu sistēmu projektēšanas uzdevums ir radīt optimālas vai tuvas optimālām (kvazioptimālas) AVS, kas nodrošina kāda optimuma kritērija (minimāls enerģijas vai izejvielu patēriņš, maksimāla rentabilitāte vai procesa norises drošums) vai vairāku kritēriju kompleksa izpildi ražošanas vidē ar mainīgiem parametriem.

Tā kā veikt organizētus eksperimentus ar dārgām tehnoloģiskām iekārtām ražošanas apstākļos ir problemātiski un reizēm neiespējami, tad tehnoloģisko procesu AVS projektēšanā **lietderīgi pielietot modelēšanas metod**i.

Varam formulēt sekojošus AVS modelēšanas mērķus un uzdevumus:

- veikt AVS posmu un to parametru optimālu izvēli;
- izveidot optimālu AVS struktūru;
- noteikt optimālu vadības algoritmu dotajam ražošanas objektam;
- izpētīt ārējo iedarbju – perturbāciju (slodzes, ražošanas vides faktoru u.c.) iespaidu uz AVS darbību;
- izstrādāt AVS atbilstoši uzdotajām kvalitātes prasībām.

Turpmāk apskatītā metodika darbam ar matemātiskās laboratorijas **MATLAB** apakšprogrammu **SIMULINK** paredzēta lauksaimniecības enerģētikas un informācijas tehnoloģiju specialitātes maģistra studiju programmu priekšmetu **„Automātisko sistēmu modelēšana"** un **"Ražošanas datorvadības sistēmas"** 1.daļas **"Tehnoloģisko procesu vadības sistēmas"** apguvei. To varēs izmantot arī citu inženierzinātņu specialitāšu maģistranti, kas bakalaura studiju programmas ietvaros **apguvuši sekojošas pamatzināšanas:**

- operatorrēķinu pamatus;
- matemātiskās modelēšanas pamatus;
- automātiskās vadības pamatus;
- datorzinību pamatus.

Pirmā nodaļa veltīta AVS modelēšanas vispārīgam apskatam. Formulēti AVS dinamiskās analīzes galvenie mērķi un uzdevumi. Apskatīta AVS struktūra, raksturīgie tipveida posmi un dinamikas vienādojumu sastādīšanas pamatprincipi.

Otrajā nodaļā apskatīta metodika darbam ar matemātiskās laboratorijas **Matlab** apakšprogrammu **„Simulink"**, kura pielietojama pārejas procesu modelēšanai tehnoloģisko iekārtu automātiskās vadības sistēmās. Aprakstīts **„Simulink"** bloku bibliotēkas saturs, algoritmisko blokshēmu sastādīšanas pamatprincipi un piemēri.

Trešajā nodaļā apskatīti tehnoloģisko procesu AVS raksturīgo komponentu (posmu) algoritmi un dinamisko pārejas procesu modelēšanas piemēri. Sastādītas AVS raksturīgo posmu un to slēgumu modelēšanas blokshēmas. Izmantojot **„Simulink"** bibliotēku un **Matlab** komandlogā sastādītu programmu, iegūtas pārejas procesu raksturlīknes dažāda veida ieejas iedarbēm.

Ceturtā nodaļa veltīta automātiskās vadības algoritmu analīzei un modelēšanai. Apskatīta elektronisko regulatoru uzbūve, darbības princips un pārejas procesa raksturlīknes dažādām ieejas iedarbēm. Doti to izvēles un parametru iestatīšanas kritēriji statiskām tehnoloģisko iekārtu vadības sistēmām ar transportkavējumu.

Piektajā nodaļā apskatīti tehnoloģisko procesu AVS modelēšanas piemēri. Veikta AVS stabilitātes un darbības kvalitātes analīze, izmantojot frekvenču kritērijus. Analizēti pasākumi AVS pārejas procesa uzlabošanai.

1. AUTOMĀTISKĀS VADĪBAS SISTĒMU MODELĒŠANAS PAMATI

Automātiskās vadības sistēmu (AVS) veido **automātiskās vadības iekārta** (AVI), **izpildiekārta** (II) un **automātiskās vadības objekts** (AVO). AVS dinamisko analīzi veic, pētot tās izturēšanos, ja tiek izjaukts stacionārais līdzsvara stāvoklis. Par pētījumu priekšmetu tādā gadījumā kļūst izejas parametra izmaiņas process laikā – tā saucamais **pārejas process** X_{iz} = f (t) pie lēcienveidīgas (kāpņveidīgas) ieejas parametra X_{ie}= f (t) izmaiņas.

Pārejas procesu izraisa iekārtu inerce (masa, siltumietilpība, induktivitāte, kapacitāte u.c.). Jo lielāka šī inerce, jo ilgāk turpinās pārejas process. Ja pārejas procesa beigās sistēma ieņem izejas stāvokli, vai arī pāriet jaunā stacionārā stāvoklī, tad saka, ka tā ir stabila. Turpretī, ja pārejas process izraisa sistēmas nepārtrauktu attālināšanos no stacionārā stāvokļa, tad sistēma ir nestabila.

AVS dinamiskās analīzes galvenie uzdevumi:

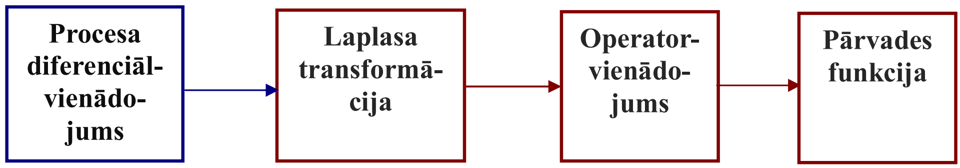
- sistēmas novērtēšana no stabilitātes viedokļa(stabila vai nestabila);
- pārejas procesa kvalitātes noskaidrošana (pārejas procesa ilgums, maksimālais pārregulējums, procesa svārstīgums u.c.);
- noteikt atsevišķo sistēmas bloku parametru iespaidu uz tās stabilitāti un pārejas procesa kvalitāti;
- noteikt ārējo iedarbju – perturbāciju iespaidu uz sistēmas stabilitāti un darbības kvalitāti.

Pārejas procesus AVS apraksta diferenciālvienādojumi vai intergro-diferenciālvienādojumi, kuru atrisinājumi parāda izejas regulējamā parametra izmaiņu laikā pie dotajām ieejas iedarbēm un perturbācijām.

Lai **sastādītu AVS dinamikas vienādojumus**, to sadala komponentēs un katrai no tām sastāda vienādojumu, kas atspoguļo tajā notiekošos fizikālos procesus. AVS komponentu dinamikas vienādojumus sastāda, izmantojot jaudas, enerģijas, masas, momenta un citu fizikālo lielumu bilances vienādojumus.

Inerciālu komponentu dinamiku parasti apraksta diferenciālvienādojumi, integrālvienādojumi vai integrodiferenciālvienādojumi, kurus algebrizē, izmantojot

Laplasa transformāciju. Pielietojot Laplasa transformāciju, iegūst algebrizētus vienādojumus, ko sauc par operatorvienādojumiem. Ņemot izejas parametra attēla attiecību pret ieejas parametra attēlu, iegūst pārvades funkciju - $\mathbf{W(s) = X_{iz}(s)/ X_{ie}(s)}$, kas ir attiecīgās komponentes darbības dinamiskais modelis, kurā mainīgie lielumi ir pārcelti no reālā laika t koordinātu sistēmas jaunā koordinātu sistēmā ar neatkarīgo mainīgo **s** kā argumentu (1.1. att).



1.1. att. **Inerciālas komponentes matemātiskā modeļa sastādīšana**

Kaut arī AVS ir ļoti daudzveidīgas, tās var sastādīt no relatīvi neliela skaita tipveida posmiem. Katra tipveida posma īpašības apraksta noteikts algoritms, pēc kura var darboties daudzas tehniskas iekārtas ar atšķirīgu uzbūvi un atšķirīgām fizikālajām īpašībām. Tas dod iespēju, izmantojot ierobežotu skaitu tipveida algoritmu, aprakstīt jebkuru AVS.

Raksturīgākie AVS tipveida posmi:

- bezinerces (proporcionālais) posms – pārejas procesu apraksta tiešās proporcionalitātes funkcija;
- pirmās kārtas inerciāls aperiodisks (statisks) posms – pārejas procesu apraksta pirmās kārtas eksponentfunkcija;
- otrās kārtas inerciāls aperiodisks (statisks) posms – pārejas procesu apraksta otrās kārtas eksponentfunkcija;
- svārstību posms – pārejas procesu apraksta rimstoša svārstīga funkcija;
- transportkavējuma posms – pārejas procesu apraksta singulāra funkcija ar kavētu argumentu;
- integrējošs posms – pārejas procesu apraksta integrālis no ieejas iedarbes;
- diferencējošs posms – pārejas procesu nosaka atvasinājums no ieejas iedarbes.

2. DINAMISKO PROCESU MODELĒŠANAS DATORPROGRAMMA *SIMULINK*

2.1. SIMULINK iespējas un īpašības

SIMULINK ir uz blokshēmām balstīta datorprogramma pārejas procesu modelēšanai dinamiskajās sistēmās. **SIMULINK** ir pilnīgi integrēta ar matemātisko aprēķinu programmu **MATLAB**. Imitējamos matemātiskos modeļus var sastādīt no standarta blokiem, kas pieejami **SIMULINK** bibliotēkā, vai no lietotāja izveidotiem blokiem.

Dažas SIMULINK īpašības

Pieejams liels skaits standarta bloku, no kuriem var sastādīt dažādu AVS simulācijas modeļus:

- **lineāru** AVS modeļus ar nepārtraukta vai diskrēta laika pārvades funkcijām;
- **nelineāru** AVS modeļus ar nepārtraukta vai diskrēta laika pārvades funkcijām;
- **loģiskās vadības** un **impulsvadības** AVS modeļus.

SIMULINK bibliotēkā ir bloki – signālu ģeneratori, kas paredzēti konstantu, kāpņveida, lineāri mainīgu, impulsveida, sinusoidālu un stohastisku (nenoteikti mainīgu) signālu formēšanai, kā arī statisko un dinamisko raksturlīkņu vizualizācijai, datu uzkrāšanai un grafiku veidošanai.

Salīdzinājumā ar matemātisko vienādojumu simulēšanas programmām, **SIMULINK** grafiskais modelis, kas tiek veidots kā algoritmiska blokshēma, ļauj viegli mainīt AVS struktūru, izmainīt saites starp blokiem vai iekļaut jaunus blokus un savienot tos ar esošajiem.

Algoritmiskā blokshēma uzskatāmi parāda AVS modeļa uzbūvi, kas savukārt ļauj padziļināt sapratni par tajā notiekošajiem procesiem.

SIMULINK dod iespēju modelēt hibrīdās AVS, kas sastāv no nepārtrauktā laika un diskrētā laika blokiem, piemēram, nepārtrauktā laika process tiek vadīts ar diskrētā laika (ciparu) kontrolleri.

Izmantojot **S-functions** (**sistēmas funkcijas**), blokus ir iespējams modelēt apakšsistēmu, kas izveidota kā algoritms vai programma, piemēram, kā analizatora vai kontrollera algoritms.

2.2. Metodiskie paskaidrojumi par SIMULINK izmantošanu

Pēc rūpīgas iepazīšanās ar aprakstu būs iespējams:

- izveidot AVS algoritmiskās blokshēmas to modelēšanai;
- modelēt AVS pārejas procesus, izvēloties simulācijas parametrus no **SIMULINK** izvēlnes;
- uzstādīt dažādus ieejas signālu un ārējo iedarbju formēšanas ģeneratorus un izejas signālu analizatorus;
- iegūt statisko raksturlīkņu vizuālos attēlus uz divkanālu plotera ekrāna;
- iegūt pārejas procesa raksturlīkņu vizuālos attēlus uz osciloskopa ekrāna;
- veikt raksturlīkņu apstrādi;
- izdrukāt AVS algoritmisko blokshēmu un saglabāt **MATLAB** failā .

Tekstā ierāmētajos laukumos ir doti darba uzdevumi. Lai labāk apgūtu programmu, visus uzdevumus ieteicams izpildīt norādītajā secībā, kā arī darba laikā izmantot tikai tās komandas, kas dotas tekstā.

Nelietojiet komandas, kas var izmainīt rezultātu vai to izdzēst!

Gaidāmais rezultāts no **SIMULINK** ir dots tālāk tekstā. Komandas, funkcijas, izteiksmes, mainīgie lielumi ir rakstīti vienkārši tekstā, liekot tos pēdiņās, piemēram, **"Save".**

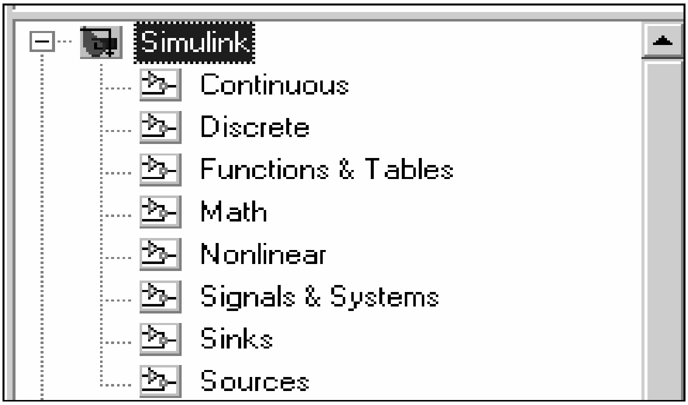
2.3. SIMULINK bibliotēkas sadaļu apraksts

2.3.1. SIMULINK bibliotēkas atvēršana

Ar peles kreiso taustiņu divreiz ieklikšķina uz ikonas **MATLAB**. Atveras **MATLAB komandlogs**. **SIMULINK** tiek atvērts ar komandu **"Simulink"** (2.1. att.).

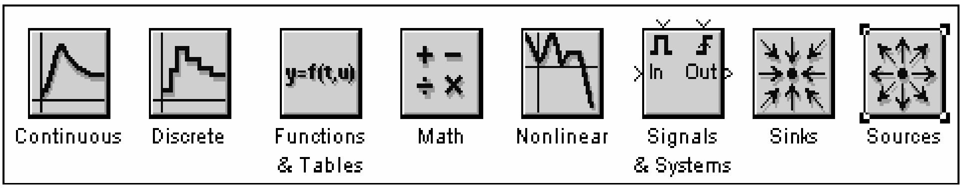
2.1. att. **SIMULINK bibliotēkas atvēršana**

Tiek atvērts **SIMULINK** bibliotēkas sadaļu saraksts, kura pamatsadaļas parādītas 2.2. attēlā. Pilnais saraksts ir ievērojami plašāks. Šeit parādām tās sadaļas, kuras galvenokārt tiek izmantotas, lai sastādītu AVS modelēšanas algoritmiskās blokshēmas. Katras sadaļas pamatbibliotēku var paplašināt ar papildblokiem no **„Simulink extras”** bibliotēkas.



2.2. att. **SIMULINK bibliotēkas pamatsadaļu saraksta logs**

Loga labajā pusē redzamas **SIMULINK** bibliotēkas sadaļu ikonas, kas personificē katras sadaļas saturu (2.3. att.).



2.3. att. **SIMULINK bibliotēkas sadaļu ikonas**

Continuous – analogie posmi (nepārtraukta laika); **Discrete** – diskrētie posmi (pārtraukta laika);

Functions & Tables – funkcijas un datu tabulas; **Math** – matemātiskās – loģiskās funkcijas;

Nonlinear – nelineārie posmi; **Signals & Systems** – signāli un sistēmas;

Sinks – signālu uztvērēji un analizatori; **Sources** – signālu avoti, ģeneratori.

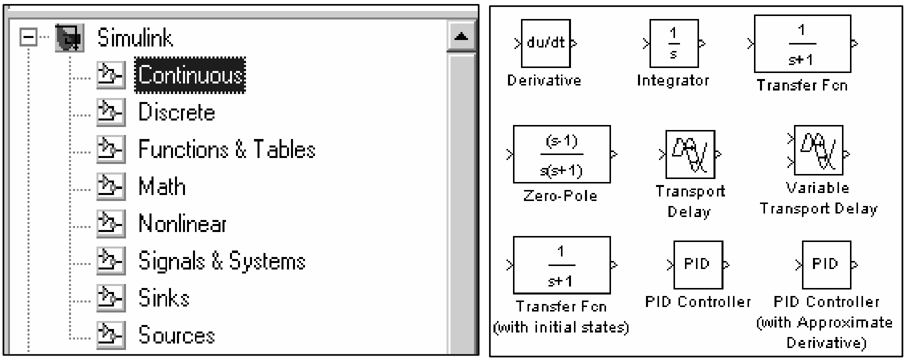
Ar dubultklikšķi uz bibliotēkas ikonas atveras neliels jauns logs, kurā parādās visi bloki šajā bibliotēkas sadaļā. Šādā veidā var detalizēti apskatīt bibliotēku (redzēt atsevišķus blokus). Pēc tam tiek parādīts bibliotēkas saturs.

Dubultklikšķis uz ikonas, lai atvērtu bibliotēkas logu. Bibliotēkas logu aizver kā parastu Windows logu (ar klikšķi uz **x**, vai izvēloties izvēlni ***File/ Close***)

Ja vēlamies detalizētāku informāciju par bloku kopā ar parametriem, kurus var mainīt, konfigurējot bloku, var veikt dubultklikšķi uz konkrētā bloka ikonas.

2.3.2. Analogo posmu bibliotēkas saturs

Analogo posmu ieejas un izejas signāli ir nepārtraukta laika funkcijas. Tos parasti apraksta pārvades funkcijas, kas iegūtas, algebrizējot posmu dinamikas vienādojumus, pielietojot nepārtraukto funkciju Laplasa transformāciju, ko sauc arī par s- transformāciju (2.4. att.).



2.4. att. **Analogo posmu bibliotēkas standartbloki**

Derivative – ieejas signāla atvasināšanas bloks (ideāls diferencējošs posms);

Integrator – integrējošs bloks (ideāls astatisks posms);

Transfer Fcn – aperiodiska (statiska) posma pārvades funkcija;

Zero-Pole – aperiodiska posma pārvades funkcija ar nullēm un poliem (par nullēm sauc posma iedarbes operatora saknes, bet par poliem – pašoperatora saknes);

Transport Delay – transportkavējuma posms ar kavētu argumentu;

Variable Transport Delay – transportkavējuma posms ar divām variējamām ieejām; **Transfer Fcn** (with initial states) – aperiodiska posma pārvades funkcija (ar nenulles sākuma nosacījumiem);

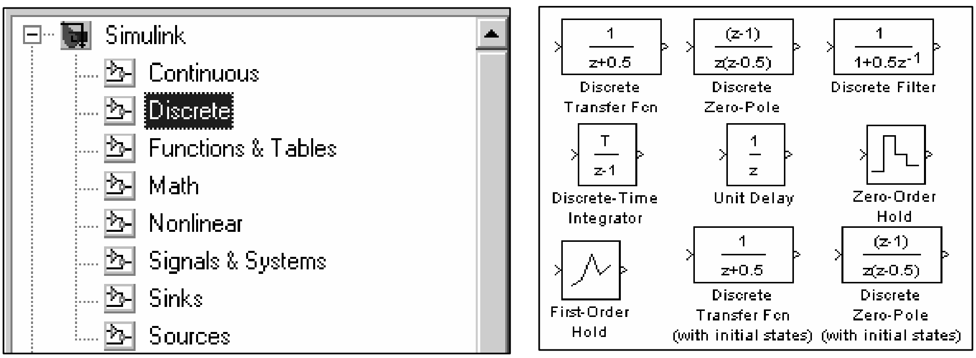
PID Controller – idealizēts PID (proporcionāli-integrālais-diferenciālais) regulators;

PID Controller (with Approximative Derivative) – Reāls PID regulators (ar aproksimētu diferencēšanas funkciju).

PID regulatoru algoritmi doti to parametru iestatīšanas logā. Taču tie ir idealizēti un pilnībā neatbilst reāla **PID regulatora** darbības algoritmam mūsdienu kontrolleros. Pieredze liecina, ka lietderīgāk ir sastādīt izvērstu **PID kontrollera** algoritmisko blokshēmu no **Simulink** standartblokiem.

2.3.3. Diskrēto posmu bibliotēkas saturs

Diskrēto posmu bibliotēkas standartblokus izmanto, lai sastādītu diskrēto (impulsu-ciparu) vadības sistēmu algoritmiskās blokshēmas un veiktu to pārejas procesu imitāciju modelēšanu Windows vidē (2.5. att.).



2.5. att. **Diskrēto posmu bibliotēkas standartbloki**

Discrete Transfer Fcn – diskrētā pārvades funkcija;

Discrete Zero-Pole – diskrētā pārvades funkcija ar nullēm un poliem;

Discrete Filter – diskrētā filtra pārvades funkcija;

Discrete – Time Integrator – diskrētā laika integrators;

Unit Delay – diskrētā transportkavējuma pārvades funkcija;

Zero-Order Hold – nulltās kārtas aizture;

First-Order Hold – pirmās kārtas aizture;

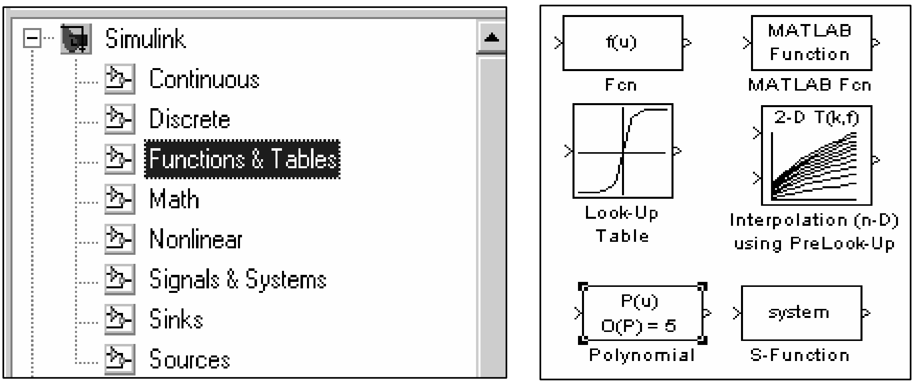
Discrete Transfer Fcn (with initial states) – diskrētā pārvades funkcija (ar nenulles sākuma nosacījumiem);

Discrete Zero-Pole (with initial states) – diskrētā pārvades funkcija ar nullēm un poliem un nenulles sākuma nosacījumiem.

Diskrēto posmu algoritmus (diskrēto pārvades funkciju veidā) iegūst sadalot nepārtraukto laiku diskrētos intervālos ar noteiktu laika soli un pielietojot diskrēto Laplasa transformāciju, ko sauc arī par z – transformāciju).

2.3.4. Funkciju un datu tabulu bibliotēkas saturs

Šī bibliotēka dod iespēju veikt signālu apstrādi, izmantojot matemātikas standartfunkcijas, polinomālas izteiksmes, sistēmfunkcijas un datu interpolācijas tabulas.



2.6. att. **Standartfunkcijas un datu tabulas**

Fcn – vispārējo funkciju bloks;

Matlab Fcn – **Matlab** funkciju bloks;

Look-Up Table – viendimensijas datu pārskata tabula;

Interpolation (n-D) using PreLook Up – daudzdimensiju datu pārskata tabula;

Polinomial – polinomālo funkciju bloks;

S-Function – sistēmfunkciju bloks.

Fcn ir universāls skaitļošanas bloks. Tajā ievadītās izteiksmes arguments ir bloka ieejas signāls, ko apzīmē ar simbolu „u”. Ja ieejas signāls ir uzdots kā vektors, tad, lai veiktu darbības ar tā atsevišķiem elementiem, konkrēti jānorāda to argumenti. Piemēram, divu ieejas signāla elementu saskaitīšanu pieraksta sekojoši: u(1) + u(2). Izteiksmju pierakstam izmanto SI – programmēšanas valodu. Izskaitļotā izteiksme attēlojas uz bloka piktogrammas.

Matlab Fcn bloks dod iespēju apstrādāt ieejas signālu, izmantojot jebkuru programmu no **Matlab** bibliotēkas vai ārpus tās , kas realizēta **M-faila** veidā.

Look-Up Table bloks nodrošina jebkuras viendimensijas funkcijas aprakstīšanu tabulārā veidā. Funkcijas grafiks, kuru apraksta iestatītie ieejas un izejas parametri, kas uzdoti diskrētu lielumu [1 2 5 8] vai nepārtraukta izmaiņas diapazona veidā [0:8.5], attēlojas uz bloka piktogrammas.

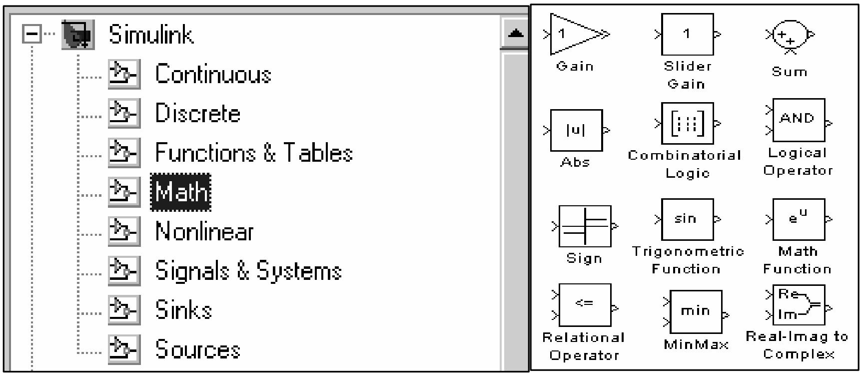
Interpolation (n-D) using PreLook Up bloks pēc darbības principa līdzīgs blokam **Look-Up Table** ar to atšķirību, ka tas dod iespēju aprakstīt daudzargumentu funkciju.

Polinomial bloks paredzēts n-tās kārtas polinomālu izteiksmju koeficientu aprēķināšanai.

S-Function ir relatīvi patstāvīga programma, kas uzrakstīta **Matlab** vai **SI** valodā un glabājas **M-failā** vai **MEX-failā** un tiek izmantota sekojošiem mērķiem:

- jaunu **Simulink** bibliotēkas bloku veidošanai;
- apakšsistēmu apraksta algoritmu veidošanai;
- animācijas līdzekļu iekļaušanai **S-modelī**.

2.3.5. Matemātisko – loģisko funkciju bibliotēkas saturs



2.7. att. **Matemātisko – loģisko funkciju standartbloki**

Gain – reizinātājs (bezinerces posms ar konstantu pārvades koeficientu);

SliderGain – slīdošais reizinātājs (bezinerces posms ar mainīgu pārvades koeficientu);

Sum – summators; **Abs** – absolūtās vērtības operators;

Combinatorial Logic – kombinatoriskā loģika;

Logical Operator – loģiskais operators;

Sign – signāla zīmes pārbaudes bloks;

Trigonometric Function – trigonometriskas funkcijas;

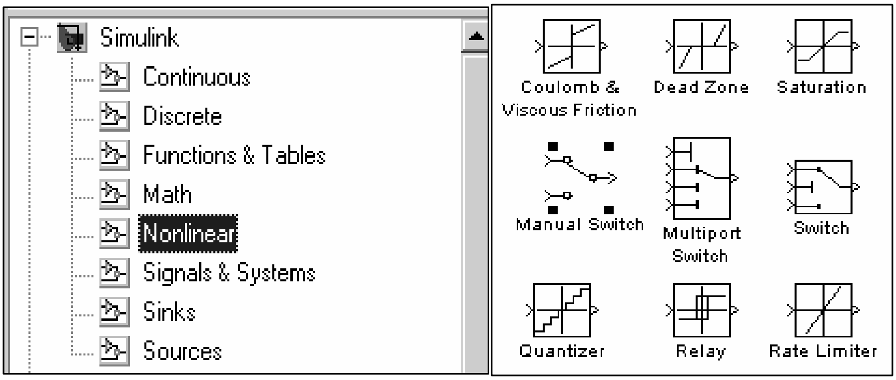
Math Function – matemātiskas netrigonometriskas funkcijas;

Relational Operator – salīdzināšanas operators;

MinMax – minimālā-maksimālā lieluma meklēšanas bloks;

Real-Imag to Complex – reāla – imagināra skaitļa pārveidošana kompleksā skaitlī

2.3.6. Nelineāro posmu bibliotēkas saturs



2.8. att. **Nelineāro posmu standartbloki**

Coulomb & Viscous Friction – Kulona spēks un viskozā berze;

Dead Zone – nejutības zona;

Saturation – signāla amplitūdas ierobežošanas bloks;

Manual Switch – divpozīciju rokas slēdzis;

Multiport Switch – daudzpozīciju automātiskais slēdzis;

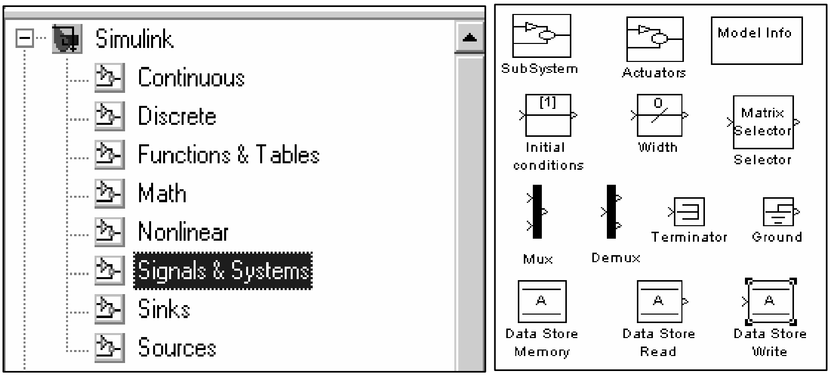
Switch – divpozīciju automātiskais slēdzis;

Quantizer – nepārtraukta signāla kvantētājs;

Relay – divpozīciju relejs;

Rate Limiter – signāla izmaiņas ātruma ierobežotājs.

2.3.7. Signālu un apakšsistēmu bibliotēkas saturs



2.9. att. **Signālu un apakšsistēmu standartbloki**

SybSystem – apakšsistēma;

Actuators – izpildmehānisma modelis;

Model Info – informācija par modeli;

Initial conditions – sākuma nosacījumi;

Width – signāla platuma noteikšana;

Selector – signālu selektors;

Mux – multipleksors (signālu jaucējs);

Demux – demultipleksors (signālu atdalītājs);

Terminator(ierobežotājs) – nepieslēgto S-modeļa izejas portu šunts;

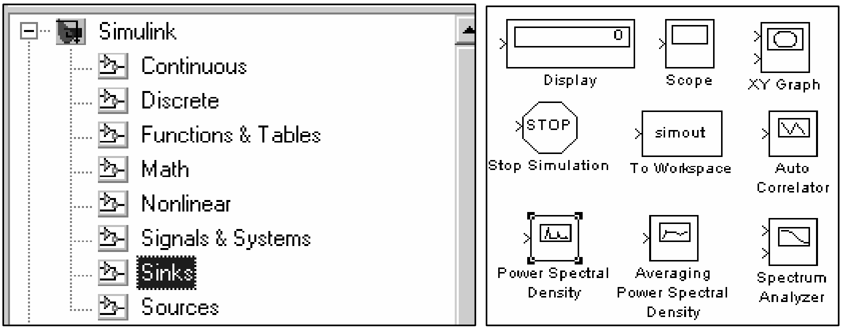
Ground (zeme) – nepieslēgto S-modeļa ieejas portu šunts;

Data Store Memory – datu atmiņa;

Data Store Read – datu nolasīšana;

Data Store Write – datu reģistrācija.

2.3.8. Signālu uztvērēju un analizatoru bibliotēkas saturs



3.10. att. **Signālu uztvērēji un analizatori**

Display – ciparu displejs aprēķinu rezultātu vizualizācijai;

Scope – osciloskops pārejas procesa raksturlīkņu vizualizācijai;

XY Graph – divkanālu ploteris statisko raksturlīkņu vizualizācijai;

Stop Simulation – modelēšanas pārtraukšanas bloks;

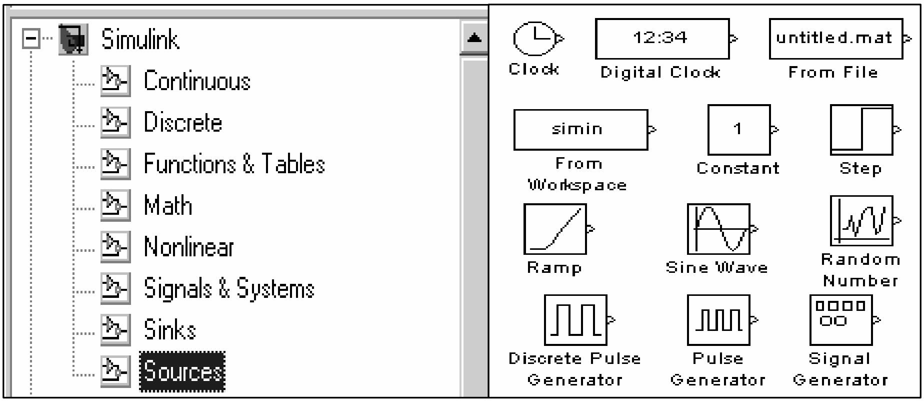
To Workspace – datu saglabāšana Matlab darba apgabalā;

Auto Corelator – autokorelators;

Power Spectral Density – signāla jaudas spektrālais blīvums;

Averaging Power Spectral Density –signāla jaudas vidējais spektrālais blīvums.

2.3.9. Signālu avotu un ģeneratoru bibliotēkas saturs



2.11. att. **Signālu avoti un ģeneratori**

Clock – nepārtraukta laika signāla avots;

Digital clock – diskrēta laika signāla avots;

From File – datu ievadīšana modelī tieši no faila;

From Workspace – datu ievadīšana modelī no Matlab darba apgabala;

Constant – konstanta signāla ģenerators;

Step – kāpņveida signāla ģenerators;

Ramp – lineāri augoša (dilstoša) signāla ģenerators;

Sine Wave – sinusoidāla signāla ģenerators;

Random Number – gadījuma skaitļu ģenerators;

Discrete Pulse Generator – pārtraukta laika impulsu ģenerators;

Pulse Generator – nepārtraukta laika impulsu ģenerators;

Signal Generator – dažādas formas signālu ģenerators.

2.4. Algoritmiskās blokshēmas sastādīšana

Mūsu uzdevums ir izpētīt AVS posmu, kuru apraksta sekojoša pārvades funkcija:

$$W(s) = \frac{X_{iz}(s)}{X_{ie}(s)} = \frac{2}{3s+1}, \qquad (2.1)$$

kur $\mathbf{X_{iz}(s)}$ – posma izejas parametra attēla funkcija;

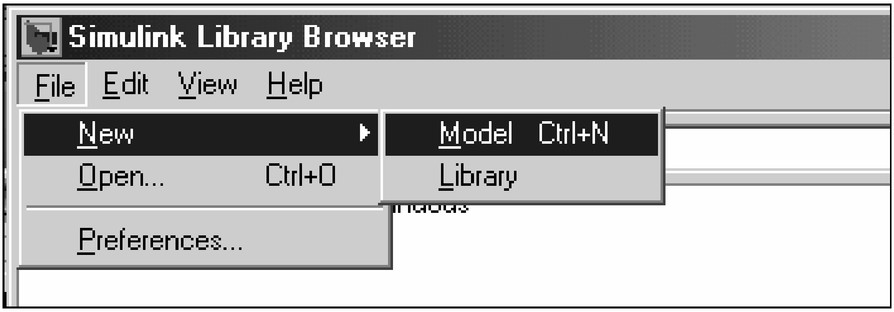
$\mathbf{X_{ie}}$ **(s)** – posma ieejas parametra attēla funkcija;

s – Laplasa arguments.

Šis ir pirmās kārtas aperiodisks posms ar **pārvades koeficientu - 2** un **laika konstanti - 3 sekundes**. Uzdodam sākuma nosacījumus: **t = 0,** $\mathbf{X_{iz}}$ **= 0.** Laika momentā **t = 0** posma ieejā padod lēcienveida iedarbi $\mathbf{X_{ie}}$ **= 4.**

Sāksim veidot modelēšanas blokshēmu ar jauna loga atvēršanu.

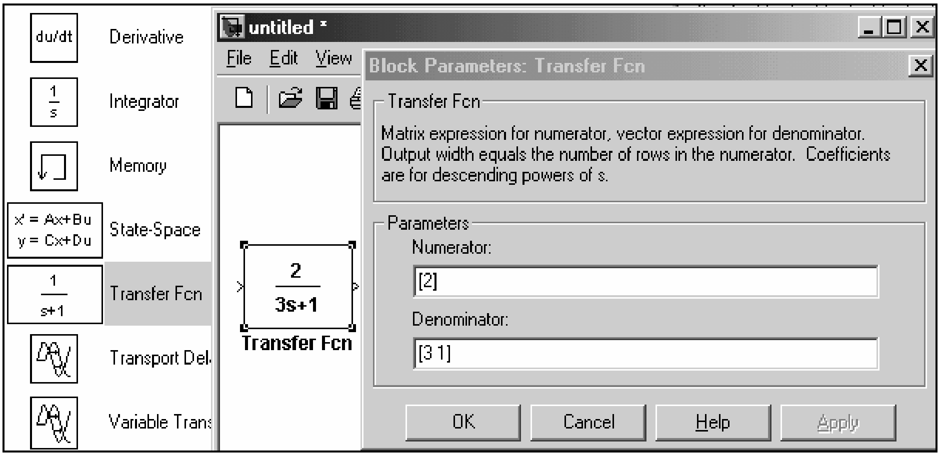
Izvēlas ***File/New/Model*** Matlab komandlogā vai ***File/New*** /***Model*** Simulink Library Browser logā



Tagad ir atvērts jauns logs blokshēmas sastādīšanai. Jaunā modeļa nosaukums sākumā ir **„untitled.”** (2.12. att.). Vēlāk, modeli ievadot Simulink failā, tā nosaukumu nomaina, piemēram, ar **„Bloks 1”.** Kad atvērts modeļa logs, tajā iekopē nepieciešamos blokus no atbilstošām Simulink bibliotēkām.

Atver **Continuous** bibliotēku. Noklikšķina ar kreiso taustiņu uz **“Transfer Fcn”** bloku. Turot nospiestu kreiso peles taustiņu, velk attiecīgo bloku uz blokshēmas **“untitled”** logu un novieto vēlamajā pozīcijā.

Noklikšķinot divas reizes ar kreiso taustiņu uz bloku **Transfer Fcn** atveras parametru logs **Block Parameters: Transfer Fcn** Blokam ir divi iestatīšanas parametri: **Numerator** – skaitītājs un **Denominator** – saucējs. Lai iestatītu pārvades funkcijas **(2.1)** parametrus, standartbloka skaitītājā nomainām koeficientu **„1”** ar koeficientu **„2”**, bet saucējā – koeficientu **„1”**pie argumenta **„s”** ar koeficientu **„3”** (2.12. att.). Kā skaitītājā, tā saucējā var ievadīt vairāku koeficientu matricu un tādā veidā izmainīt pašu algoritmu. Piemēram, ievadot saucējā **[3 2 1]**, iegūst sekojošu izteiksmi: $\mathbf{3s^2 + 2s + 1}$.



2.12. att. **Bloka Transfer Fcn parametru iestatīšana**

Lai veiktu pārejas procesa simulāciju, bloka ieejā padod organizētu signālu. Parasti tas ir kāpņveida (lēcienveida) signāls, ko formē ar bloku **Step** no signālu avotu un ģeneratoru bibliotēkas (2.11. att.).

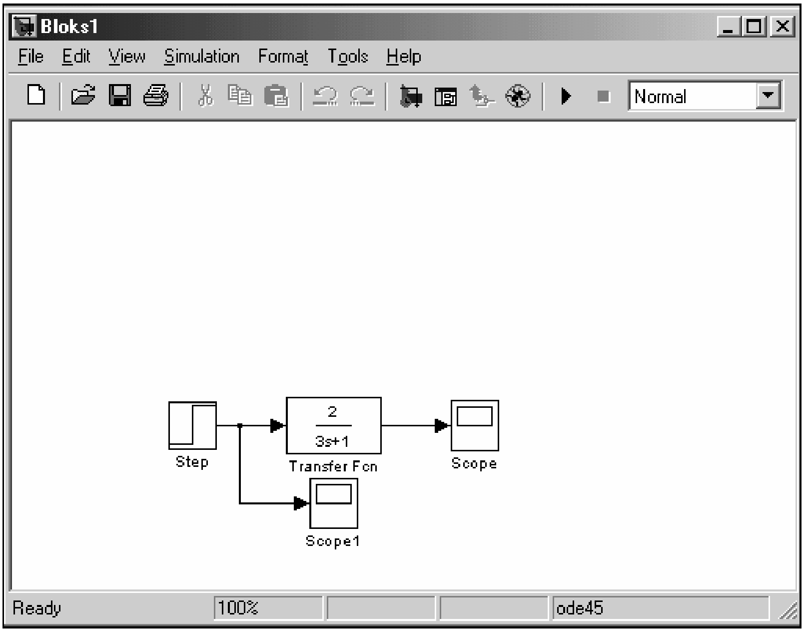
Atver **Sources** bibliotēku. Noklikšķina ar kreiso taustiņu uz **„Step”** bloku. Turot nospiestu kreiso peles taustiņu, velk attiecīgo bloku uz blokshēmas **„untitled”** logu un novieto vēlamajā pozīcijā.

Lai iegūtu pētāmā bloka pārejas procesa raksturlīkni $\mathbf{X_{iz} = f(t)}$, tā izejā uzstāda osciloskopu.

Atver **Sinks** bibliotēku. Noklikšķina ar kreiso taustiņu uz **„Scope”** bloku. Turot nospiestu kreiso peles taustiņu, velk attiecīgo bloku uz blokshēmas **„untitled”** logu un novieto vēlamajā pozīcijā.

Lai savienotu blokus ar saitēm, novieto kursoru uz bloka sākumu vai beigām. Kursors pārvēršas par ”+”. Turot nospiestu kreiso peles taustiņu, velk līniju uz otra bloka beigām vai sākumu līdz kursors atkal pārvēršas par „+”. Atlaižot kreiso taustiņu, parādās sakaru līnija ar bultu, kas savieno abus blokus. Kad blokshēma pilnībā izveidota, to ievadām **Matlab** direktorijā kā **mdl. failu**.

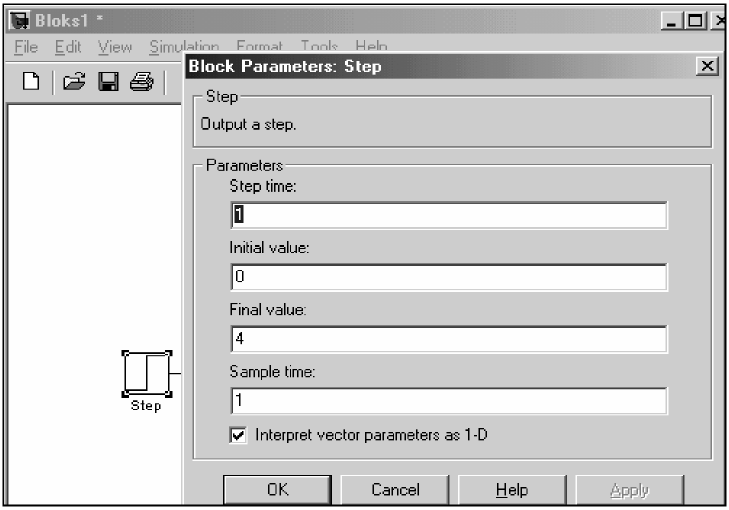
Izvēlne ***File/Save As*** blokshēmas logā ***„Untitled”*** ievada **„Bloks 1”** Blokshēma tiek saglabāta **Matlab** direktorijā. Šī direktorija tiek parādīta pēc komandas ***„Save As”.*** Rezultāts redzams 2.13. attēlā.



2.13. att. **Modelēšanas blokshēma**

Tagad nepieciešams konfigurēt ieejas signāla formēšanas bloku **„Step”.**

Ar kreisā peles taustiņa dubultklikšķi atver **„Step”** bloku . Tad parādās parametru iestatīšanas logs. Uzstāda parametrus: **„Step time”** (laika solis) = **1**; **„Initial value”** (signāla lielums laika soļa sākumā) = **0**; **„Final value”** (signāla lielums laika soļa beigās) = **4**; **„Sample time”** (laika etalons - saskaņošanas laiks ar bloku **„Transfer Fcn**) = **1**. Ievadot visas vērtības, šo logu aizver apstiprinot to ar **Apply – OK** (2.14. att.).



2.14. att. **Bloka Step konfigurēšana**

Bloka **„Scope”** konfigurācijas parametrus iestatīsim simulācijas beigās.

Tagad apskatīsim modelēšanas procesa sagatavošanu un norisi.

2.5. Pārejas procesa modelēšana no izvēlnes

Pieņemsim, ka blokshēma **„Bloks”**, kura parādīta 2.13. attēlā ir atvērta. Ja tā nav atvērta, tad to atver ,izmantojot **MATLAB** komandrindu.

2.5.1. Modelēšanas parametru iestatīšana

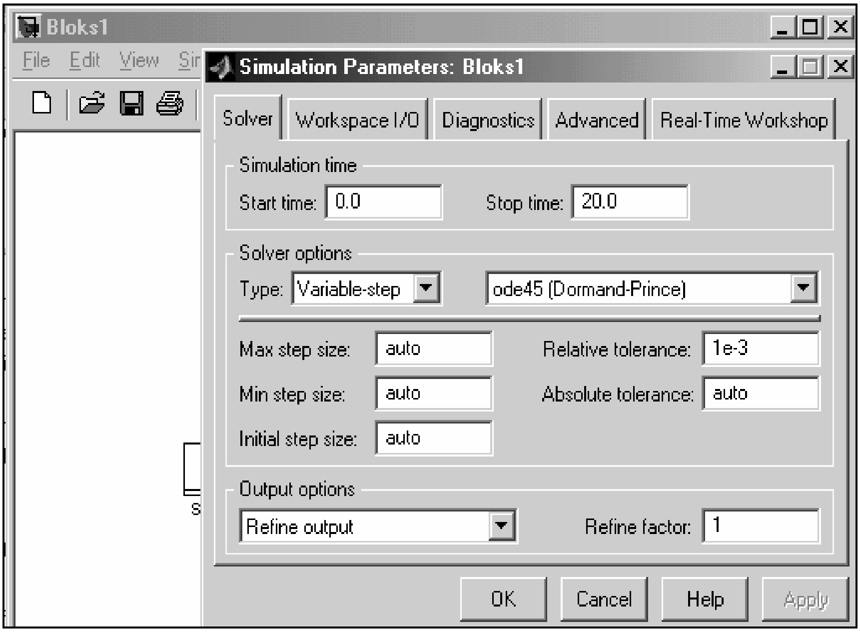
Pirms modelēšanas jāiestata daži parametri:

Lieto izvēlni ***Simulation/Parameters/Solver*** blokshēmas logā **„Bloks1”.** Atveras parametru izvēles logs **Simulation Parameters: Bloks 1** (2.15. att.).

Tiek piedāvātas dažas opcijas:

- modelēšanas „**Start time**” (starta laiku) un „**Stop tim**e” (beigu laiku) izvēlas atbilstoši pārejas procesa ilgumam (lielāku vai vienādu ar procesa laiku);
- lietojot „**Solver options**” (atrisinājuma opcija), tiek piedāvāts izmantot „**Variable-ste**p” (mainīgā soļa) vai „**Fixed-step**” (fiksētā soļa) metodi;

- nepārtraukta laika sistēmām kā likums izmanto „**Variable-ste**p” jeb Dormanda – Prince metodi, diskrētā laika sistēmām – „**Fixed-step**”.



2.15. att. **Modelēšanas parametru izvēle**

Sistēmas pārejas procesa analīzei izmanto „**ode 45**” metodi, kuru sauc arī par „Runge – Kutta” ceturto metodi. Izmantojot mainīgā soļa metodi, sākuma soļa **„Initial step size**” un maksimālā soļa **„Max step size”** optimālos parametrus **„auto”** režīmā **Simulink** iestata automātiski. Šis optimālais laika solis var tikt izmainīts modelēšanas laikā.

Apskatīsim dažas citas izvēlnes ***Simulation/parameters*** opcijas. „**Workspace I/O**” (darba apgabala ieeja/izeja) dod iespēju operēt ar modelēšanas datiem un rezultātiem, kas izvietoti **Matlab** darba apgabalā. **„Limit data points”** definē laukuma lielumu, kurā tiek attēlotas aprēķinātās vērtībās. Tam jāpiešķir pietiekami daudz vietas. Ja tas nav izdarīts, tiek saglabāta tikai daļa no simulācijas rezultāta. Ja uzstādītais punktu skaits ir **1000**, tad drošāk izvēlēties lielāku, piemēram, **5000**.

2.5.2. Modelēšanas procedūra un rezultāti

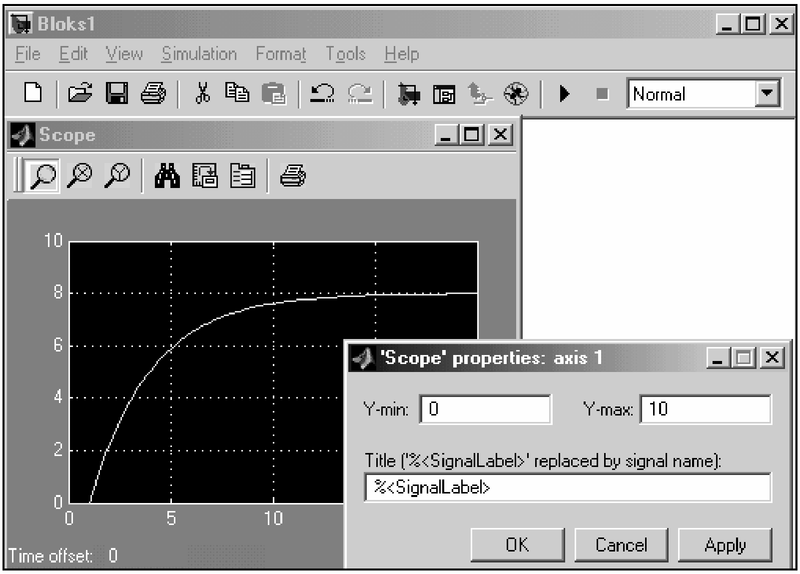
Pirms modelēšanas iestata sākuma un beigu laikus. Sākuma laiku parasti izvēlas vienādu ar **„0”**. Beigu laika izvēli nosaka pētāmās ierīces inerce. Inerciāliem posmiem ieteicams izvēlēties simulācijas beigu laiku lielāku par **5** laika konstantēm.

Piemērā dotajam posmam (**pārvades funkcija 2.1**) laika konstante **T = 3 s**.

Izvēlnē **Simulation/ Parameters** iestata „**start time**” = 0 un **„stop time”** = 20 sek., un apstiprina ar **Apply – OK**. Pārējos parametrus atstāj nemainīgus. Aizver izvēlnes logu. Sāk modelēšanu. Izvēlas **Simulation/Start** blokshēmas logā „**Bloks1**”. Ja simulācija jāpārtrauc, noklikšķina **Simulation/Stop.**

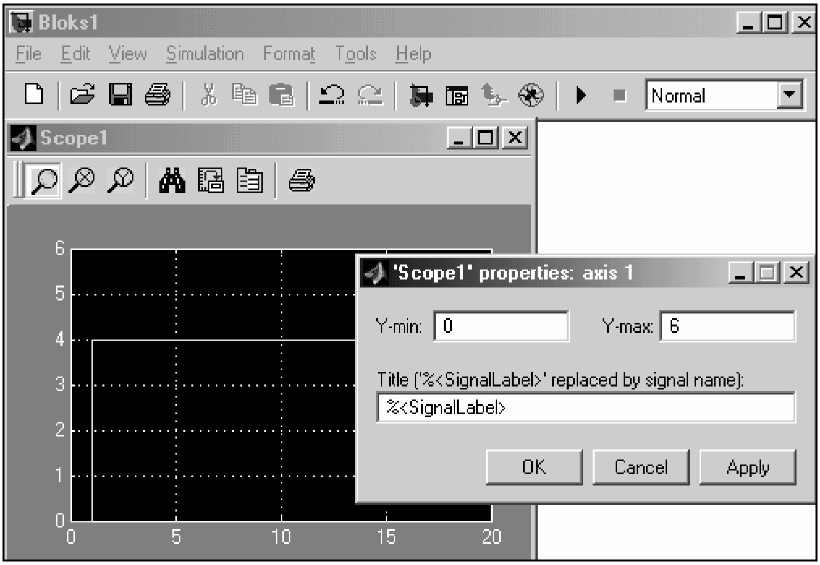
Modelēšanas rezultāts **X_{iz} = f (t),** ja **X_{ie} = 0, t < 1 un X_{ie} = 4, t ≥ 1,** redzams logā **„Scope”** (2.16. att.), kuru atver divreiz noklikšķinot uz bloka **Scope**.

Lai koriģētu **X_{iz}** ass parametrus, noklikšķina ar peles labo taustiņu uz raksturlīknes laukumu. Parādās opciju logs. Ar kreiso taustiņu noklikšķina **Axes properties.** Atveras logs – „**Scope” properties: axis 1.** Tajā ir divu parametru – **Y-min** un **Y-max** iestatīšanas logi. Iestatām **Y-min = 0, Y-max = 10 → Apply-OK.** Automātiski pārstatās Y-ass parametri.



2.16. att. **Pētāmā bloka pārejas procesa raksturlīkne X_{iz} = f(t), $X_{iz\,max}$ = 8**

Lai apskatītu ieejas signāla raksturlīkni $\mathbf{X_{ie}}$ **= f(t)**, ar kreiso peles taustiņu noklikšķina uz bloka „**Scope 1**”. Atveras logs **Scope 1,** kurā redzama $\mathbf{X_{ie}}$ **= f(t)** raksturlīkne (2.17. att.). Nepieciešamības gadījumā var koriģēt $\mathbf{X_{ie}}$ **-ass** parametrus līdzīgi, kā tas tika darīts iepriekš.



2.17. att. **Ieejas signāla** $\mathbf{X_{ie}}$ **= f(t) raksturlīkne,** $\mathbf{X_{ie\ max}}$ **= 4**

Dotajā piemērā tika izmantoti divi atsevišķi osciloskopi **„Scope”** un **„Scope 1”** (2.13. att.). Lai iegūtu ieejas un izejas lielumu pārejas procesa raksturlīknes $\mathbf{X_{ie}}$**= =f(t)** un $\mathbf{X_{iz}}$ **= f(t)** vienā oscilogrammā, var izmantot vienu osciloskopu, kuram uzstāda divas ieejas.

Automātikas pamatu kursā noskaidrojām, ka jebkuru AVS var sadalīt komponentēs, kuru statiskās un dinamiskās īpašības apraksta noteikti algoritmi. AVS komponentes, kuru darbību apraksta viens un tas pats algoritms sauc par AVS raksturīgo posmu. Zinot kāds raksturīgais posms apraksta to vai citu AVS komponenti, var veikt šīs komponentes statisko un dinamisko īpašību imitāciju modelēšanu Windows vidē, izmantojot **Matlab** apakšprogrammu **„Simulink”.** AVS raksturīgāko komponentu modelēšanas piemēri apskatīti nākošajā nodaļā.

3. TEHNOLOĢISKO PROCESU AVS RAKSTURĪGO POSMU MODELĒŠANA

Raksturīgākie tipveida posmi, kuros var sadalīt tehnoloģisko procesu automātiskās vadības sistēmas ir sekojoši:

❐ **bezinerces – proporcionālais posms** (elektriska ķēde ar aktīvo pretestību, operacionālais pastiprinātājs ar aktīvajiem elementiem, tahoģenerators inerciālā sistēmā, mehāniskais pārvads bez slīdes un brīvgājiena, frekvenču pārveidotājs inerciālā sistēmā, vielas plūsmas regulēšanas droseļvārsts);

❐ **pirmās kārtas aperiodisks – statisks – inerciāls posms** (laika releji, taimeri, elektriskie sildelementi, laika aiztures ķēdes ar aktīvo pretestību un induktivitāti vai kapacitāti, tahoģenerators un frekvenču pārveidotājs mazinerciālā sistēmā, mehānismu piedziņas elektrodzinēji, elektroģeneratori, temperatūras mērīšanas pārveidotāji, ēku apsildes katli, siltumapgādes objekti, ūdensapgādes objekti, iekšdedzes dzinēji, elektriskie hidrauliskie vai pneimatiskie izpildmehānismi ar cietu atgriezenisko saiti);

❐ **otrās kārtas aperiodisks – statisks – inerciāls posms** (divpakāpju RC laika aiztures ķēde, elektriskā žāvēšanas kamera ar masīvu sildķermeni, tvaika katls ar divfāžu vidi (ūdens + tvaiks), divu tilpumu irigācijas slūžu kaskāde, ķīmiskie reaktori, kas veido savienotos traukus);

❐ **svārstību posms** (rotācijas ātruma centrbēdzes regulators ar svārstību slāpētāju; RLC radiotehniskais svārstību kontūrs, šķidruma līmeņa kontroles pludiņš, mehāniskais svārsts, solenoida elektromagnētiskais vārsts ar atsperi);

❐ **transportkavējuma posms** (materiālu un detaļu pneimotransporta iekārtas, robottehnisko kompleksu transporta moduļi, siltuma, gāzes un šķidruma pārvades cauruļvadi, plūsmas līniju transportieri un konveijeri);

❐ **integrējošs posms** (elektriskie, hidrauliskie un pneimatiskie izpildmehānismi bez atgriezeniskās saites tvaika apgādes, ūdensapgādes vai siltuma apgādes inženiersistēmās, integratori, integrālie regulatori, astatiskie vadības objekti);

❐ **diferencējošs posms** (diferencējošais transformators, diferencējoša CR ķēde, tahoģenerators, ja tā ieejas lielums ir vārpstas pagrieziena leņķis, diferenciālais žiroskops, AVS virknes korekcijas ķēde ar diferencējošo filtru).

3.1. Frekvenču pārveidotāja kā bezinerces posma pielietošana

Frekvenču pārveidotāju var aprakstīt ar bezinerces posmu pie nosacījuma, ka tā laika konstante **T** ir daudzkārt mazāka par vadības objekta laika konstanti $\mathbf{T_{obj}}$ **($\mathbf{T \ll T_{obj}}$).** Lai modelis būtu adekvāts reālajai sistēmai, nosacītā bezinerces posma laika konstantei jāapmierina sekojošs nosacījums: $\mathbf{T \leq 0.1\ T_{obj}}$.

Frekvenču pārveidotājā inerci rada elektriskie un elektromagnētiskie pārejas procesi elektromagnētiskās saderības, līdzstrāvas un motora filtros. Atkarībā no komplektācijas, frekvenču pārveidotāja laika konstante $\mathbf{T_f \leq 0.05}$ **s.** To nedrīkst ignorēt vadot mazinerciālus objektus, piemēram, zāģmašīnu, kas sastāv no padeves mehānisma ar frekvenču regulējamu asinhrono piedziņu un zāģu agregāta kā vadības objekta ar laika konstanti $\mathbf{T_{obj} \leq 0.2}$ **s**. Tā kā šī laika konstante ir salīdzināma ar frekvenču pārveidotāja laika konstanti tad zāģmašīnas vadības sistēmā frekvenču pārveidotājs jāapskata kā inerciāls posms.

Atšķirīga situācija ir lielinerciālās sistēmās, piemēram, notekūdeņu aerācijas kompresoru ar frekvenču regulējamo asinhrono piedziņu vadības sistēmās. Vadības objekts ir notekūdeņu aerācijas tvertne, kurā gaisa skābekļa pārneses procesa inerci nosaka laika konstante $\mathbf{T_{obj} \geq 10}$ **min**., kas ir daudzkārt lielāka par frekvenču pārveidotāja laika konstanti. Šādā sistēmā frekvenču pārveidotājs darbosies kā ideāls bezinerces posms. Tad var pieņemt, ka padodot frekvenču pārveidotāja ieejā lēcienveida iedarbi (vadības līdzspriegumu **U**), arī izejas lielums (maiņsprieguma frekvence **f**) mainīsies lēcienveidīgi bez aizkavēšanās. Pie kam **f** izmainīsies tieši proporcionāli **U**.

Frekvenču pārveidotāja kā bezinerces posma dinamisko pārejas procesu reālā laika koordinātu sistēmā apraksta **oriģinālvienādojums**:

$$f(t) = K \cdot U(t), \tag{3.1}$$

kur **f(t)** – izejas sprieguma frekvence kā laika funkcija, $\mathbf{s^{-1}}$;

U(t) –ieejas spriegums kā laika funkcija, **V**;

K= f/ U – statiskais pārvades koeficients, $\mathbf{s^{-1}}$ **/V** ;

f un **U** – izejas un ieejas lielumu statiskā nostabilizējusies vērtība pārejas procesa beigās.

Pielietojot Laplasa transformāciju, no (3.1) iegūst **operatorvienādojumu**:

$$f(s) = K \cdot U(s), \tag{3.2}$$

kur **f(s)** – izejas sprieguma frekvences attēls;

U(s) –ieejas sprieguma attēls;

s – Laplasa arguments.

No operatorvienādojuma (3.2) iegūst frekvenču pārveidotāja **pārvades funkciju** (dinamisko pastiprinājuma koeficientu):

$$W = f(s)/ U(s) = K. \tag{3.3}$$

Kā redzams frekvenču pārveidotāja īpašības kā statiskā, tā dinamiskā režīmā nosaka pārvades koeficients **K**, kurš raksturo ierīces jutību pret ieejas spriegumu **U**.

3.2. Frekvenču pārveidotāja kā bezinerces posma modelēšana

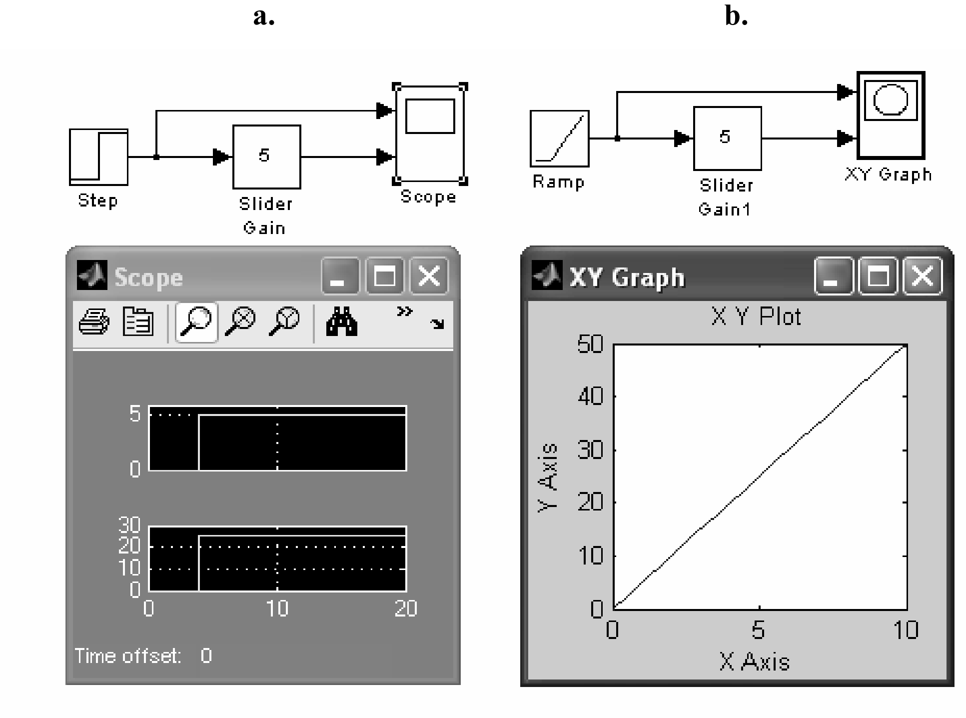
Sastādām modelēšanas blokshēmu (3.1.a att.), kas sastāv no **„Simulink”** bibliotēkas standartblokiem: **„Step”, „Slider gain”** un **„Scope”**.

Frekvenču pārveidotāja ieejā padodam vadības spriegumu: **U = (0 ÷ 10) V**. Proporcionāli ieejas spriegumam mainās izejas sprieguma frekvence: **f = (0 ÷50) s^{-1} (herci)**. Sakarību starp frekvenču pārveidotāja izejas un ieejas lielumiem kā statiskā, tā dinamiskā režīmā izsaka pārvades koeficients **K = f/U = 50/10 = 5 s^{-1}/V**. Mainoties vadības spriegumam par **1V**, izejas sprieguma frekvence mainās par **5 s^{-1}**.

Konfigurējam blokus **„Step”, „Slider gain”** un **„Scope”**. Bloks **„Step”** formē lēcienveida ieejas spriegumu. Iestatām brīvi izvēlētu sākuma laiku, piemēram, **t_0 = 4 s** un spriegumu **U = 5V**. Atveram bloku **„Slider gain”,** kurš modelē frekvenču pārveidotāju, un ar slīdni iestatām **K = 5 s^{-1}/V**. Veicot simulāciju, iegūstam ieejas un izejas lieluma pārejas procesa raksturlīknes: **U(t)** un **f(t)** (3.1.a att.).

Lai modelētu frekvenču pārveidotāja statisko raksturlīkni **f(U)**, ieejā uzstāda lineāra signāla ģeneratoru **„Ramp”**, bet izejā – divkanālu ploteri **„XY Graph”**. Veicot simulāciju, iegūstam statisko raksturlīkni, kas parāda lineāro sakarību starp frekvenču pārveidotāja ieejas spriegumu **U (X – ass)** un izejas sprieguma frekvenci **f (Y – ass)**

(3.1.b att.). Redzam, ka frekvenču pārveidotāja statiskā raksturlīkne ir lineāra, kas liecina, ka tā jutība nav atkarīga no ieejas signāla.



3.1. att. **Frekvenču pārveidotāja modelēšanas blokshēmas un raksturlīknes: a – pārejas procesa raksturlīknes U(t), f(t); b – statiskā raksturlīkne f(U)**

3.3. Pirmās kārtas statisku inerciālu posmu modelēšanas piemēri

Pirmās kārtas inerciāls posms ir plaši izplatīts. Tas apraksta dinamiskos pārejas procesus elektroniskajās, elektriskajās, elektromagnētiskajās, termodinamiskajās, mehāniskajās, hidrauliskajās un pneimatiskajās iekārtās. Padodot posma ieejā lēcienveida iedarbi, izejas lielums izmainās aperiodiski pēc eksponenciāla likuma. Tas izskaidrojums ar pārejas procesa inerci, kuras lielumu raksturo laika konstante **T**. Jo lielāka laika konstante, jo lēnāk noris pārejas process, kura beigās izejas lielums nostabilizējas un sasniedz noteiktu statisku vērtību. Automātikas pamatu kursā noskaidrojām, ka pirmās kārtas inerciālu statisku posmu apraksta pirmās kārtas diferenciālvienādojums ar konstantiem koeficientiem - pārvades koeficientu **K**, kas nosaka posma jutību pret ieejas iedarbi un laika konstanti T, kura nosaka pārejas procesa inerci. Turpmāk apskatīsim šādu posmu modelēšanas piemērus.

3.3.1. Elektriska sildelementa statiskās un dinamiskās īpašības

1. Elektriska sildelementa ieejas iedarbe ir elektriskā jauda P

Elektriskais sildelements ir tipisks pirmās kārtas inerciāls statisks posms, kura silšanas procesu apraksta homogēns **pirmās kārtas diferenciālvienādojums** ar konstantiem koeficientiem:

$$T\frac{d\tau}{dt}+\tau=K_p\cdot P, \tag{3.4}$$

kur **T = c·m/ (α ·S)** – sildelementa silšanas laika konstante, **s**;

$\mathbf{K_p = 1/(\alpha \cdot S)}$ – sildelementa statiskais pārvades koeficients pēc jaudas, $\mathbf{^oC/W}$;

$\mathbf{\tau = (\theta - \theta_0)}$ – sildelementa virstemperatūra (**θ** – virsmas temperatūra, $\mathbf{\theta_0}$ – sākuma temperatūra aukstam stāvoklim), $\mathbf{^oC}$.

P – pievadītā elektriskā jauda, **W**;

c – sildelementa īpatnējā siltumietilpība, $\mathbf{J/(kg\cdot {}^oC)}$;

m – sildelementa masa, **kg**;

α – siltuma atdeves koeficients, $\mathbf{W/(m^2\cdot {}^oC)}$;

S – sildelementa virsmas laukums, $\mathbf{m^2}$;

Redzam, ka elektrisku sildelementu apraksta pirmās kārtas inerciāls statisks posms.

Pielietojot Laplasa transformāciju, iegūstam sildelementa **operatorvienādojumu** un **pārvades funkciju**:

$$T\cdot\tau(s)\cdot s+\tau(s)=K_p\cdot P(s); \tag{3.5}$$

$$W_p(s)=\frac{\tau(s)}{P(s)}=\frac{K_p}{T\cdot s+1}. \tag{3.6}$$

Siltuma pārejas procesa modelēšanai apskatīsim konkrētu piemēru ar skaitliski uzdotiem vai aprēķinātiem sildelementa fizikālajiem parametriem.

Sildelements ar nominālo jaudu $\mathbf{P_{nom} = 500\ W}$ un spriegumu $\mathbf{U_{nom} = 220\ V}$ ir daudzslāņu cilindrs ar lineāru siltuma avotu – nihroma sildspirāli, kas ievietota tērauda korpusā un izolēta ar presēta kristāliskā magnija oksīda slāni.

Sildelementa garums **l = 1 m** un diametrs **d = 13 mm**. Aprēķinātais virsmas laukums: $\mathbf{S = \pi \cdot d \cdot l = 3.14 \cdot 13 \cdot 10^{-3} \cdot 1 = 0.04\ m^2}$**.**

Siltuma atdeves koeficients pie gaisa piespiedu cirkulācijas ātruma **v = 1.5 m/s** ir $\mathbf{\alpha = 29\ W/(m^2 \cdot {}^oC)}$**.** Tad aprēķinātais sildelementa pārvades koeficients pēc jaudas: $\mathbf{K_p = 1/(\alpha \cdot S) = 1/(29 \cdot 0.04) = 0.84\ {}^oC/W}$**.**

Sildelementa silšanas laika konstanti var aprēķināt, izmantojot tuvinātu formulu:

$$T = \frac{c_v \cdot m}{\alpha \cdot S}, \tag{3.7}$$

kur $\mathbf{c_v = 580\ J/(kg\ {}^oC)}$ – sildelementa vidējā īpatnējā siltumietilpība;

m = 0.67 kg – sildelementa masa.

No (3.7) iegūstam, ka laika konstante $T = \dfrac{580 \cdot 0.67}{29 \cdot 0.04} = 335\ s = 5.6 \min$.

Ievietojot izteiksmē (3.6) konstanšu skaitliskos lielumus, iegūstam sildelementa pārvades funkcijas analītisko izteiksmi:

$$W_p(s) = \frac{0.84}{5.6 \cdot s + 1} \tag{3.8}$$

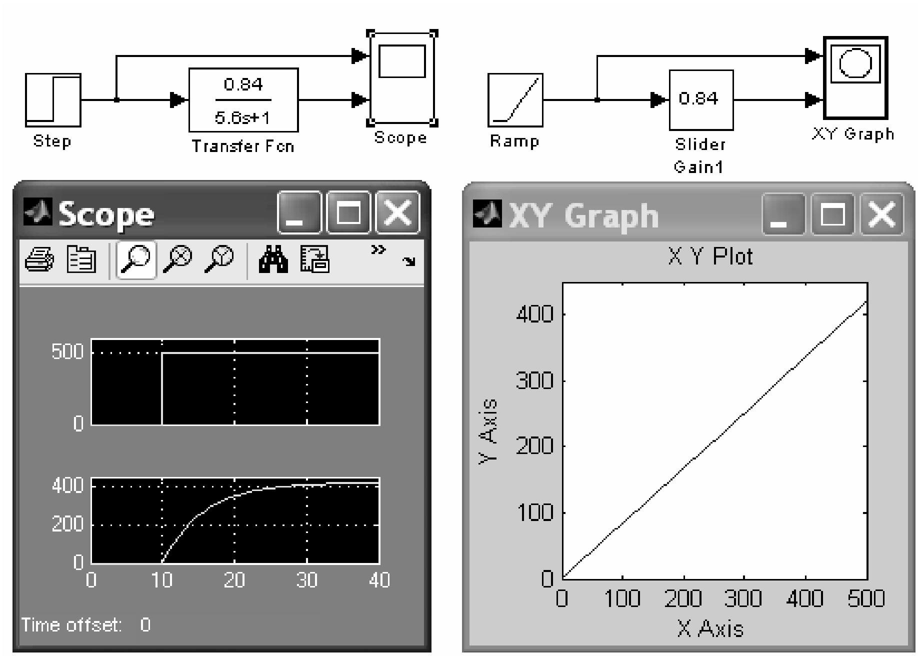
Modelēšanas blokshēmas un raksturlīknes. Lai iegūtu elektriska sildelementa pārejas procesa raksturlīknes, sastādām modelēšanas blokshēmu (3.2.a att.), kas sastāv no **„Simulink”** bibliotēkas standarta blokiem: **„Step”**, **„Transfer Function”** un **„Scope”.**

Pieņemsim, ka laika momentā $\mathbf{t_0 = 10\ min}$. sildelements tiek pieslēgts spriegumam $\mathbf{U_{nom} = 220\ V}$ un tajā izdalās elektriskā jauda $\mathbf{P_{nom} = 500\ W}$, kas visa pārvēršas siltumā. Notiek pārejas process, kura beigās sildelementa virstemperatūra sasniedz maksimālo nostabilizējušos lielumu $\mathbf{\tau_s = K_p \cdot P = 0.84 \cdot 500 = 420\ {}^oC}$**.**

Konfigurējam blokus **„Step”, „Transfer Function”** un **„Scope”.** Bloks **„Step”** formē lēcienveida ieejas jaudu. Iestatām modelēšanas sākuma laiku $\mathbf{t_0}$ **= 10 min**. un jaudu $\mathbf{P_{nom}}$ **= 500 W**. *Laika t_0 izvēle nav reglamentēta, taču, lai iegūtu uzskatāmāku modelēšanas rezultātu, izvēlas $t_0>0$.*

Bloks **„Transfer Function”** modelē sildelementa silšanas dinamisko procesu. Atveram bloku un iestatām $\mathbf{K_p = 0.84}$ **°C/W** un **T = 5.6 min**. Lai iegūtu pilnu pārejas procesu, simulācijas laiku izvēlas sekojošu: $\mathbf{t_{sim} > t_0 + 4T}$.

a. **b.**



3.2. att. **Sildelementa modelēšanas blokshēmas un raksturlīknes, ja ieejas iedarbe ir jauda P (W) un izejas lielums – virstemperatūra τ (°C): a – pārejas procesa raksturlīknes P = f(t), τ = f(t); b–statiskā raksturlīkne τ_s = f(P)**

Veicot simulāciju, iegūst sildelementa ieejas lieluma - jaudas un izejas lieluma – virstemperatūras pārejas procesa raksturlīknes: **P = f(t)** un **τ = f(t)** (3.2.a att.).

Noteiksim sildelementa nostabilizējušos temperatūru **θ**, ja aukstā stāvoklī pirms elektriskās jaudas pieslēgšanas tā ir $\mathbf{\theta_0 = 15}$ **°C:** $\mathbf{\theta = \tau_s + \theta_0 = 420 + 20 = 440}$ **°C.**

Lai modelētu sildelementa statisko raksturlīkni $\mathbf{\tau_s}$ **= f(P)**, ieejā uzstāda lineāra signāla ģeneratoru **„Ramp”**, bet izejā – divkanālu ploteri **„XY Graph”.**

Statisko sakarību **τ_s = f(P)** modelē ar bloku **„Slider gain**”, kurā ievadām **K_p = 0.84 $^oC/W$.** Konfigurējam bloku **„Ramp”**, ievadot tajā sākuma jaudu **P_0 = 0** un beigu jaudu **P_b = P_{nom} = 500 W**, kā arī jaudas izmaiņas ātrumu **20 W/min**.

Veicot simulāciju, iegūst raksturlīkni, kas parāda sakarību starp nostabilizējušos virstemperatūru **τ_s (Y–ass)** un atbilstošo elektrisko jaudu **P (X–ass)** (3.2.b att.).

<u>**2. Elektriska sildelementa ieejas iedarbe ir spriegums U**</u>

Ja kā regulējamu ieejas iedarbi izmanto spriegumu U, tad elektriska sildelementa silšanas dinamisko procesu apraksta **nelineārs diferenciālvienādojums**:

$$T\frac{d\tau}{dt} + \tau = K_U \cdot U^2 , \tag{3.9}$$

kur **K_U = 1/(α·S·R)** – sildelementa pārvades koeficients pēc sprieguma, **$^oC/V^2$**;

R = U_{nom}^2/P_{nom} = 220^2/500 = 97 Ω – nihroma sildspirāles nominālā elektriskā pretestība.

Matlab „Simulink” tehnoloģija dod iespēju modelēt nelineārus nestacionārus procesus bez statisko raksturlīkņu linearizācijas. Apskatīsim sekojošu vienkāršotu paņēmienu. Pieņemsim, ka sildelementa ieejas iedarbe nav spriegums **U**, bet tā kvadrāts – **U^2**. Ievedam jaunu mainīgo: **υ = U^2, V^2**. Tad iegūstam **lineāru diferenciālvienādojumu** ar konstantiem koeficientiem:

$$T \cdot \frac{d\tau}{dt} + \tau = K_U \cdot \nu , \tag{3.10}$$

kur **T = 5.6 min.** – sildelementa silšanas laika konstante;

K_U = 1/(α·S·R) = 1/(29·0.04·97) = 0.0087 $^oC/V^2$ – pārvades koeficients.

Pielietojot Laplasa transformāciju, no **(3.10)** iegūstam **pārvades funkciju**:

$$W_U(s) = \frac{\tau(s)}{\nu(s)} = \frac{K_U}{T \cdot s + 1} = \frac{0.0087}{5.6 \cdot s + 1} , \tag{3.11}$$

kur **τ(s)** – sildelementa virstemperatūras attēls;

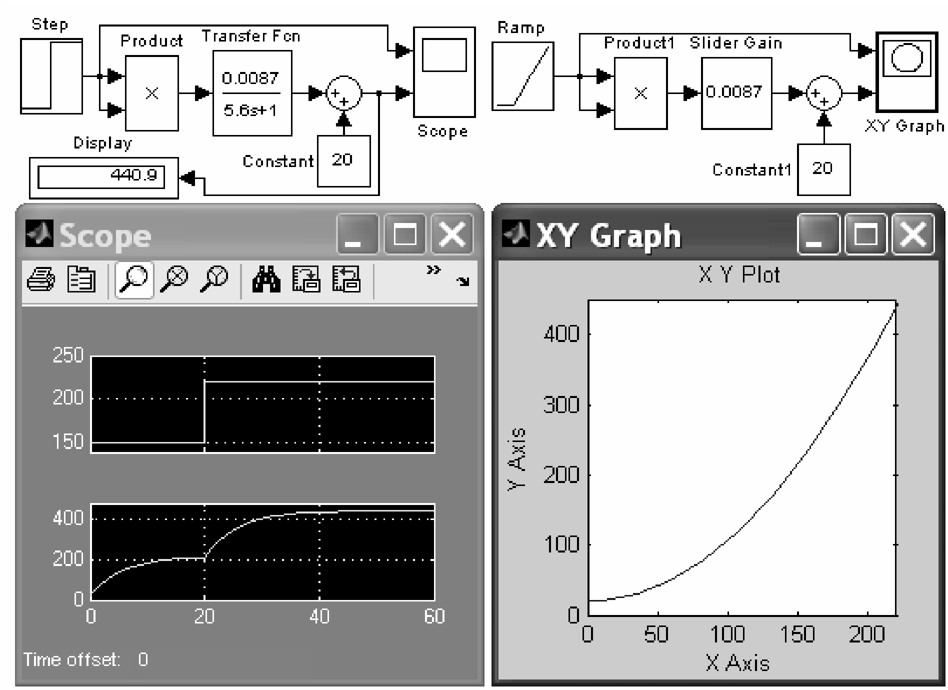
υ(s) – sprieguma kvadrāta attēls.

Modelēšanas blokshēmas un raksturlīknes. Lai iegūtu elektriska sildelementa pārejas procesa raksturlīknes, sastādām modelēšanas blokshēmu (3.3.a att.), kas sastāv no **„Simulink”** bibliotēkas standarta blokiem: **„Step”, „Product”, „Transfer Function”, „Sum”, „Constant”, „Display”** un **„Scope”**.

Bloks **„Product”** realizē reizināšanas funkciju, kas nepieciešama, lai iegūtu sildelementa ieejas lieluma – sprieguma kvadrātu kā divu spriegumu reizinājumu:

$$v = U^2 = U \times U . \tag{3.12}$$

a. **b.**



3.3. att. **Modelēšanas blokshēmas un raksturlīknes, ja ieejas iedarbe ir spriegums U (V) un izejas lielums – temperatūra θ (^{0}C): a – pārejas procesa raksturlīknes U = f(t), θ = f(t); b – statiskā raksturlīkne θ = f(U)**

Modelēsim divus silšanas pārejas procesus pie diviem dažādiem spriegumiem. Pieņemsim, ka laika momentā $\mathbf{t_0 = 0}$ sildelements ar sākuma temperatūru $\mathbf{\theta_0 = 20\ ^oC}$ tiek pieslēgts spriegumam $\mathbf{U_1 = 150\ V}$. Notiek pārejas process, kura beigās sildelementa virstemperatūra sasniedz nostabilizējušos lielumu – $\boldsymbol{\tau_{s1} = K_U \cdot U_1^2 = 0.0087 \cdot 150^2 = 196\ ^oC}$ un temperatūra – $\boldsymbol{\theta_{\tau 1} = 196 + 20 = 216\ ^oC}$.

Kad pirmais pārejas process beidzies, laika momentā $\boldsymbol{t_1 = 20}$ **min** spriegumu lēcienveidīgi palielinām līdz nominālajam spriegumam $\boldsymbol{U_2 = 220}$ **V**. Notiek pārejas process, kura beigās sildelementa virstemperatūra sasniedz lielumu – $\boldsymbol{\tau_{s2} = K_U \cdot U_2^2 = 0.0087 \cdot 220^2 = 421\ ^oC}$ un temperatūra – $\boldsymbol{\theta_{\tau 2} = 421 + 20 = 441\ ^oC}$.

Konfigurējam blokus **„Step”**, **„Transfer function”, ”Constant”** un **„Scope”**. Atveram bloku **„Step”** un iestatām sākuma spriegumu $\boldsymbol{U_1 = 150}$ **V**, beigu spriegumu $\boldsymbol{U_2 = 220}$ **V** un otrās sprieguma pakāpes ieslēgšanas laiku $\boldsymbol{t_1 = 20}$ **min**., kas ir pietiekams, lai beigtos pirmais pārejas process. Atveram bloku **„Transfer function”** un iestatām sildelementa pārvades koeficientu $\boldsymbol{K_U = 0.0087\ ^oC/V^2}$ un laika konstanti $\boldsymbol{T = 5.6}$ **min.** Atveram bloku **„Constant”** un iestatām sildelementa sākuma temperatūru $\boldsymbol{\theta_0 = 20\ ^oC}$. Atveram logu **„Simulation parameters”** un izvēlamies simulācijas kopējo laiku $\boldsymbol{t_{sim} = 60}$ **min**, kas ir pietiekams, lai sildelementa temperatūra pilnībā nostabilizētos.

Veicot simulāciju, uz osciloskopa **„Scope”** ekrāna iegūstam sildelementa ieejas lieluma - sprieguma un izejas lieluma – temperatūras pārejas procesa raksturlīknes: $\boldsymbol{U = f(t)}$ un $\boldsymbol{\theta = f(t)}$ (3.3.a att.), kur **t** – laiks, min. Bloks **„Display”** fiksē izejas lieluma - temperatūras maksimālo nostabilizējušos skaitlisko vērtību.

Lai iegūtu sildelementa statisko raksturlīkni $\boldsymbol{\theta = f(U)}$, ieejā uzstādām lineāra signāla ģeneratoru **„Ramp”**, bet izejā – divkanālu ploteri **„XY Graph”** (3.3.b att.). Sprieguma kvadrātu $\boldsymbol{\upsilon = U^2}$ iegūstam ar bloku **„Product1”**, bet statisko sakarību $\boldsymbol{\tau_s = f(U)}$ ar bloku **„Slider gain”**, kurā ievadām $\boldsymbol{K_U = 0.0087\ ^oC/V^2}$. Lai iegūtu statisko raksturlīkni $\boldsymbol{\theta_s = f(U)}$, uzstādām bloku **„Constant1”** un summatoru **„Sum”**.

Konfigurējam bloku **„Ramp”**, ievadot tajā sākuma spriegumu $\boldsymbol{U_0 = 0}$ un sākuma laiku $\boldsymbol{t_0 = 0}$, kā arī sprieguma augšanas ātrumu – **10 V/min**. Sprieguma augšanas ātruma izvēli nosaka simulācijas laiks, lai tajā tiktu uzņemta pilna statiskā raksturlīkne.

Veicot simulāciju, iegūstam statisko raksturlīkni, kas parāda sakarību starp sildelementa ieejas spriegumu **U (X – ass)** un nostabilizējušos temperatūru $\boldsymbol{\theta_s}$ **(Y – ass)**

(3.3.b att.). Redzam, ka dotajā gadījumā elektriskā sildelementa statiskā raksturlīkne $\boldsymbol{\theta_s = f(U)}$ ir nelineāra, jo to apraksta nelineāra funkcija: $\boldsymbol{\theta_s = \theta_0 + K_U \cdot U^2}$.

3.3.2. Rezervuāra kā ūdens līmeņa regulēšanas objekta modelēšana

Apskatīsim ūdens rezervuāru kā automātiskās vadības objektu, kura ieejas iedarbe ir ūdens plūsma **Q**, kuru sūknis caur ventiļiem padod uz rezervuāru. Tā izejas lielums ir ūdens līmenis **H**, kuru iespaido mainīga perturbācija – patēriņa plūsma $\boldsymbol{Q_p}$ (3.4. att.). Sākumā atvērti rokas vadības ventiļi $\boldsymbol{V_1}$ un $\boldsymbol{V_3}$.

Vadības objekts ir stacionārā līdzsvara stāvoklī, ja $\boldsymbol{H = H_0} =$ **const.**, $\boldsymbol{Q = Q_0} =$ **const**. un $\boldsymbol{Q_p = Q_{p0}} =$ **const.** Stacionārs stāvoklis iestājas pie nosacījuma, ka ūdens patēriņš $\boldsymbol{Q_{p0}}$ ir vienāds pieplūdi $\boldsymbol{Q_0}$.

Ieslēdzot elektromagnētisko ventili $\boldsymbol{V_2}$, lēcienveidīgi tiek izjaukts plūsmu līdzsvars kā rezultātā ūdens līmenis sāk paaugstināties. Paaugstinoties līmenim, palielinās ūdens spiediens, kas savukārt izraisa patēriņa plūsmas palielināšanos. Sakarā ar procesa inerci, notiek pakāpeniska ūdens līmeņa $\boldsymbol{H = H_0 + \Delta H(t)}$ paaugstināšanās , kamēr iestājas jauns stacionārs līdzsvara stāvoklis: $\boldsymbol{H = H_0 + +\Delta H(\infty)}$, kur $\boldsymbol{\Delta H(\infty)}$ – nostabilizējusies (maksimālā) līmeņa izmaiņa.

Dinamisko ūdens līmeņa izmaiņas procesu rezervuārā apraksta pirmās kārtas **diferenciālvienādojums** ar konstantiem koeficientiem:

$$T_{obj}\frac{d\Delta H}{dt} + \Delta H = K_{obj}\Delta Q, \qquad (3.13)$$

kur $\boldsymbol{d\Delta H/dt}$ – ūdens līmeņa izmaiņas ātrums, **m/s**;

$\boldsymbol{\Delta Q}$ – ūdens pieplūdes lēcienveida izmaiņa, $\boldsymbol{m^3/s}$;

$K_{obj} = \Delta H(\infty)/\Delta Q$ – rezervuāra statiskais pārvades koeficients, $\boldsymbol{m/(m^3/s)}$;

$T_{obj} = S \cdot \Delta H(\infty)/\Delta Q$ – rezervuāra laika konstante, **s**

S – rezervuāra šķērsgriezuma laukums, $\boldsymbol{m^2}$.

Statiskais pārvades koeficients $\boldsymbol{K_{obj}}$ izsaka objekta jutību pret ieejas iedarbi $\boldsymbol{\Delta Q}$. Jo lielāka līmeņa izmaiņa pie vienas un tās pašas ieejas iedarbes, jo objekts ir jutīgāks. Laika konstante $\boldsymbol{T_{obj}}$ raksturo līmeņa nostabilizēšanās inerci. Tā ir tieši proporcionāla ūdens līmeņa regulēšanas tilpumam (**V = S · ΔH (∞)**), bet apgriezti proporcionāla ūdens patēriņa plūsmas izmaiņai statiskā režīmā (**$\Delta Q_p = \Delta Q$**).

Aprakstītajam objektam piemīt pašizlīdzināšanās spēja, t.i., katram noteiktam ūdens pieplūdes daudzumam **Q** atbilst noteikts nostabilizējies līmenis **H**. Pie šāda nosacījuma rezervuārs ir statisks inerciāls līmeņa regulēšanas objekts (3.4. att.).

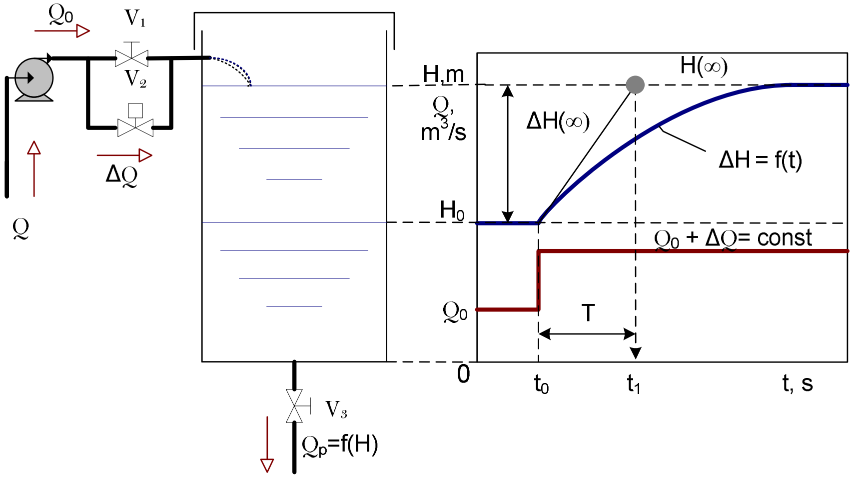
Pielietojot Laplasa transformāciju pie nulles sākuma nosacījumiem (ja **t = 0**, tad **ΔH = 0** un **ΔQ = 0),** no diferenciālvienādojuma (3.13) iegūstam ūdens rezervuāra **operatorvienādojumu** un **pārvades funkciju**:

$$T_{obj} \cdot \Delta H(s) \cdot s + \Delta H(s) = K_{obj} \cdot \Delta Q(s)\ ; \tag{3.14}$$

$$W_{obj}(s) = \frac{\Delta H(s)}{\Delta Q(s)} = \frac{K_{obj}}{T_{obj} \cdot s + 1}, \tag{3.15}$$

kur **ΔH(s) –** ūdens līmeņa izmaiņas attēls;

ΔQ(s) – ūdens pieplūdes izmaiņas attēls.



3.4. att. **Rezervuāra kā ūdens līmeņa regulēšanas objekta pārejas procesa raksturlīknes mainīgam patēriņam Q_p = f(H)**

Ūdens līmeņa pārejas procesa modelēšanai apskatīsim konkrētu piemēru ar skaitliski uzdotiem vai aprēķinātiem rezervuāra fizikālajiem parametriem.

Statiskā stāvoklī ūdens līmenis rezervuārā ar šķērsgriezuma laukumu $\mathbf{S = 2\ m^2}$ ir konstants ($\mathbf{H = H_0 = 3m}$) un ūdens plūsmas ir līdzsvarā ($\mathbf{Q_0 = Q_{p0}}$). Ieslēdzot elektromagnētisko ventili $\mathbf{V_2}$, lēcienveidīgi tiek palielināta ūdens pieplūde par lielumu $\mathbf{\Delta Q = 0.03\ m^3/s}$, kā rezultātā ūdens līmenis paaugstinās par $\mathbf{\Delta H(\infty) = 0.9\ m}$.

<u>Aprēķinātie lielumi</u>: $K_{obj} = \Delta H(\infty)/\Delta Q = 0.9/0.03 = 30 m/(m^3/s);$

$$T_{obj} = S \cdot \Delta H(\infty)/\Delta Q = S \cdot K_{obj} = 2 \cdot 30 = 60 s.$$

<u>Modelēšanas blokshēmas un raksturlīknes</u>. Lai iegūtu rezervuāra kā ūdens līmeņa **H** regulēšanas objekta pārejas procesa raksturlīknes, sastādām modelēšanas blokshēmu, kas sastāv no **„Simulink”** bibliotēkas standarta blokiem: **„Step”, „Transfer Function”, „Sum”, „Constant”, „Display”** un **„Scope”** (3.5.a att.).

Konfigurējam blokus **„Step”**, **„Transfer function”, ”Constant”** un **„Scope”**. Atveram bloku **„Step”** un iestatām plūsmas izmaiņu $\mathbf{\Delta Q = 0.03\ m^3/s}$. Atveram bloku **„Transfer function”** un iestatām rezervuāra pārvades koeficientu $\mathbf{K_{obj} = 30\ m/(m^3/s)}$ un laika konstanti $\mathbf{T_{obj} = 60\ s}$. Atveram bloku **”Constant”** un iestatām uzdoto sākuma līmeni $\mathbf{H_0 = 3\ m}$.

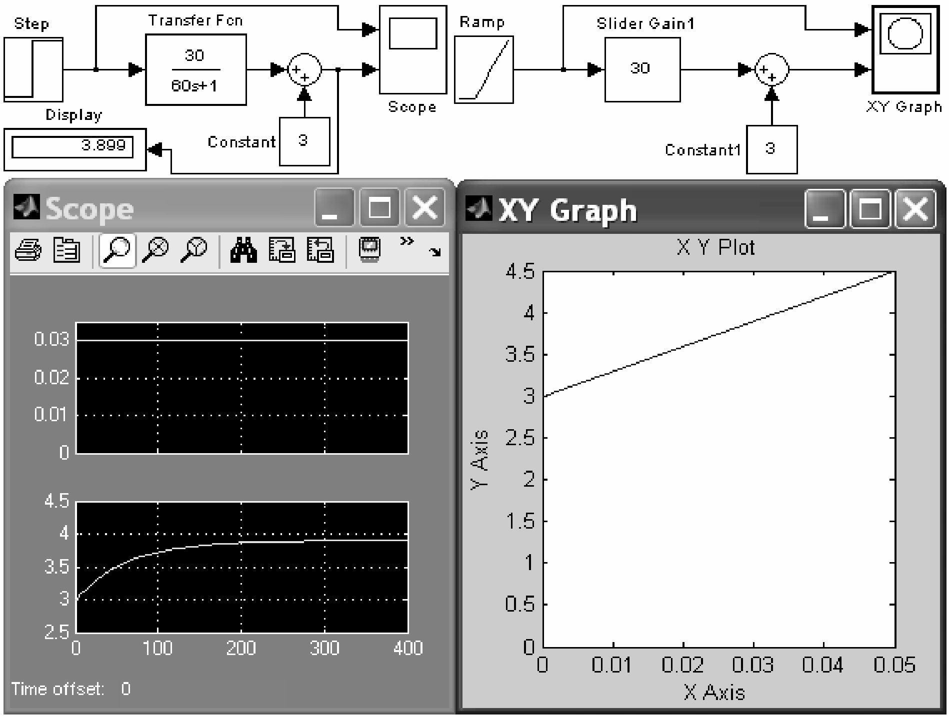
Atveram logu **„Simulation parameters”** un izvēlamies simulācijas kopējo laiku $\mathbf{t_{sim} = 400\ s}$, kas ir pietiekams, lai ūdens līmenis pilnībā nostabilizētos.

Veicot simulāciju, uz osciloskopa **„Scope”** ekrāna iegūstam rezervuāra ieejas lieluma – pievadītās ūdens plūsmas izmaiņas $\mathbf{\Delta Q}$ un izejas lieluma – ūdens līmeņa **H** pārejas procesa raksturlīknes: $\mathbf{\Delta Q = f(t)}$ un **H = f(t) (3.5.a.att.).** Bloks **„Display”** fiksē līmeņa **H** maksimālo nostabilizējušos skaitlisko vērtību - $\mathbf{H(\infty) = H_0 + \Delta H(\infty) = H_0 +}$ $\mathbf{Kobj \cdot \Delta Q}$. Ievietojot skaitliskos lielumus, iegūstam aprēķināto maksimālo līmeni: $\mathbf{H(\infty) = H_0 + K_{obj} \cdot \Delta Q = 3 + 30 \cdot 0.03 = 3.9\ m.}$

Lai iegūtu rezervuāra statisko raksturlīkni, sastādām modelēšanas blokshēmu, kuras ieejā uzstāda lineāra signāla ģeneratoru **„Ramp”**, bet izejā – divkanālu ploteri **„XY Graph”** (3.5.b att.). Statisko sakarību $\mathbf{\Delta H = f(\Delta Q)}$ modelē ar bloku **„Slider**

gain", kurā ievada $\mathbf{K_{obj} = 30\ m/(m^3/s)}$. Lai iegūtu statisko sakarību $\mathbf{H(\infty) = f(\Delta Q)}$, uzstādām konstanta signāla ģeneratoru **„Constant1"** un summatoru **„Sum"**.

a. **b.**



3.5. att. **Ūdens rezervuāra dinamisko un statisko raksturojumu modelēšanas blokshēmas un raksturlīknes, ja ieejas iedarbe ir plūsmas lēcienveida izmaiņa ΔQ (m^3/s) un izejas lielums – līmenis H (m): a – pārejas procesa raksturlīknes ΔQ = f(t), H = f(t); b – statiskā raksturlīkne H (∞) = f(ΔQ)**

Konfigurējam bloku **„Ramp"**, ievadot tajā līmeņa augšanas ātrumu, piemēram, **0.1 m/s**. Līmeņa augšanas ātruma izvēli nosaka simulācijas laiks, lai tajā tiktu uzņemta pilna statiskā raksturlīkne.

Konfigurējam bloku **„Constant1"**, ievadot tajā sākuma līmeni $\mathbf{H_0 = 3\ m}$**,** kā arī bloku **„XY Graph",** uzstādot ūdens plūsmas izmaiņas sākuma un beigu vērtības uz **X – ass** **(ΔQ = 0 – 0.05 m^3/s)** un ūdens līmeņa sākuma un beigu vērtības uz **Y – ass** **(H(∞) = 0 – 4.5 m).**

Veicot simulāciju, iegūstam statisko raksturlīkni, kas parāda sakarību starp ūdens rezervuāra ieejas iedarbi **ΔQ** un nostabilizējušos līmeni **H(∞)** (3.3.b att.). Ierobežotā parametru izmaiņas apgabalā ūdens līmenis rezervuārā mainās lineāri atkarībā no ūdens plūsmas izmaiņas, jo to apraksta linearizēta funkcija Δ **H(∞) =** $\mathbf{K_{obj}}$ **·ΔQ**.

3.4. Žāvēšanas kameras kā otrās kārtas statiska posma modelēšana

Otrās kārtas inerciāls posms parasti apraksta divu tilpumu (masu) statiskus tehnoloģiskos objektus, piemēram, zāģmateriālu žāvētavu ar gaisa masu un žāgmateriālu masu, tvaika katlu ar tvaika masu un ūdens masu, divu tilpumu ķīmiskos reaktorus, kuri darbojas pēc savienoto trauku principa. Dinamiskos pārejas procesus šādās iekārtās apraksta otrās kārtas diferenciālvienādojums ar konstantiem koeficientiem. Šādu AVS posmu sauc arī par otrās kārtas aperiodisku posmu, jo tā pārejas process ir monotons (aperiodisks).

Apskatīsim elektrisku žāvēšanas kameru kā divu tilpumu (masu) automātiskās vadības objektu, kurš sastāv no sildķermeņa un sildāmā gaisa (3.6. att.). Sastādot siltuma plūsmu bilances vienādojumus sildķermenim un gaisa masai, iegūstam žāvēšanas kameras diferenciālvienādojumu. Tā sastādīšanas metodika detalizēti tika apskatīta automātikas pamatos. Tādēļ šeit to atkārtosim saīsinātā veidā.

1. Vispirms izvēlamies mainīgos lielumus un sākuma nosacījumus:

$$\theta_g - \theta_0 = \tau_g = 0;\ \theta - \theta_0 = \tau = 0;\ P = 0, \qquad (3.16)$$

kur $\boldsymbol{\theta_g}$ – gaisa temperatūra žāvēšanas kamerā, $^{\mathbf{o}}$**C**;

$\boldsymbol{\theta}$ – sildķermeņa virsmas temperatūra, $^{\mathbf{o}}$**C**;

$\boldsymbol{\theta_0}$ – vides temperatūra, $^{\mathbf{o}}$**C**;

P – sildspirāles elektriskā jauda, **W**.

2. Sastādām elektriskā sildķermeņa siltuma plūsmu bilances vienādojumu nestacionāram režīmam:

$$Q_a + Q = P\ , \tag{3.17}$$

kur $Q_a = C \cdot m \cdot \dfrac{d\tau}{dt}$ – sildķermenī akumulētā siltuma plūsma, **W**;

Q $= \alpha \cdot S \cdot \tau$ – sildķermeņa atdotā siltuma plūsma, **W**;

$\boldsymbol{\alpha_s}$ – siltuma atdeves koeficients, $\mathbf{W/(m^2 \cdot {}^0C)}$;

$\mathbf{S_s}$ – sildķermeņa virsmas laukums, $\mathbf{m^2}$.

3. Sastādām gaisa masas siltuma plūsmu bilances vienādojumu nestacionāram režīmam :

$$Q_{ag} + Q_z = Q\ , \tag{3.18}$$

kur $Q_{ag} = C_g \cdot m_g \cdot \dfrac{d\tau_g}{dt}$ – gaisa masā akumulētā siltuma plūsma, **W**;

$Q_z = \alpha_z \cdot S_z \cdot \tau_g$ – siltuma zudumu plūsma caur kameras sienām, **W**;

$Q = \alpha \cdot S \cdot (\tau - \tau_g)$ – kameras gaisa masas saņemtā siltuma plūsma, **W**;

$\boldsymbol{\alpha_z}$ – siltuma zudumu koeficients, $\mathbf{W/(m^2 \cdot {}^oC)}$;

$\mathbf{S_z}$ – siltuma zudumu virsmas laukums , $\mathbf{m^2}$;

$\mathbf{c_g}$ – gaisa īpatnējā siltumietilpība, $\mathbf{J/(kg \cdot {}^oC)}$;

$\mathbf{m_g}$ – kameras gaisa masa, **kg**.

No izvērsto vienādojumu (3.17) un (3.18) sistēmas, iegūstam žāvēšanas kameras **diferenciālvienādojumu**, kas apraksta gaisa masas virstemperatūras pārejas procesu atkarībā no kameras parametriem un sildķermenim pievadītās elektriskās jaudas:

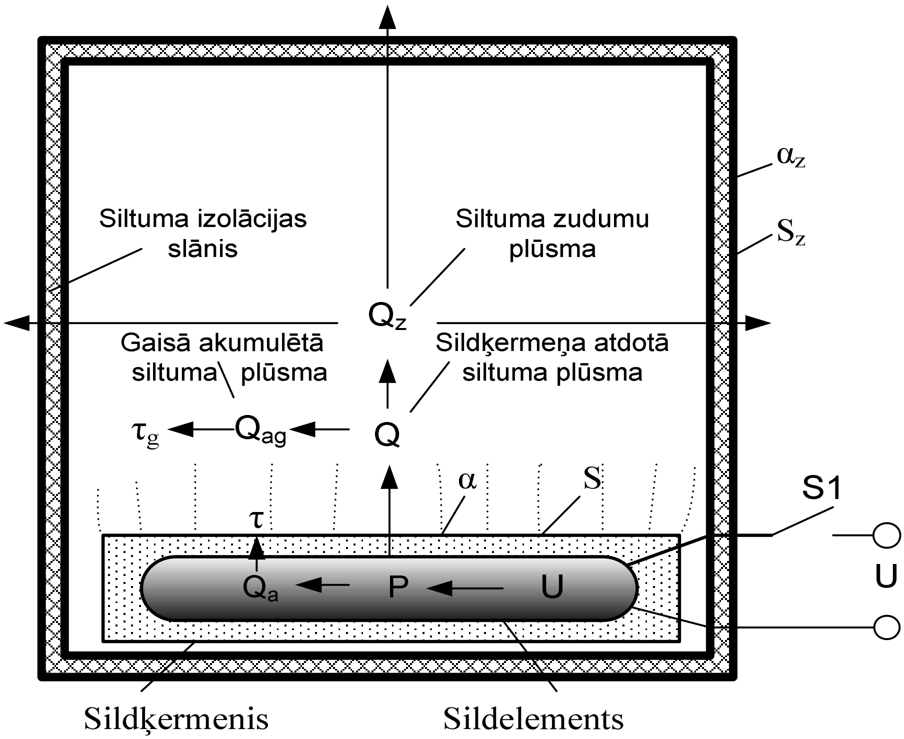
$$T_1^2 \frac{d^2\tau_g}{dt^2} + T_2 \frac{d\tau_g}{dt} + \tau_g = K_{obj} \cdot P\ , \tag{\textbf{3.19}}$$

kur $K_{obj} = 1/(\alpha \cdot S + \alpha_z \cdot S_z)$ – kameras siltuma pārvades koeficients, $\mathbf{{}^oC/W}$;

$T_1 = \sqrt{T \cdot T_g}$ *un* $T_2 = T + T_g$ – siltuma pārvades inerces laika konstantes, **s**.

$\mathbf{T_g = (c_g \cdot m_g)/(\alpha \cdot S + \alpha_z \cdot S_z)}$ – kameras gaisa silšanas laika konstante, **s**;

$\mathbf{T = (c \cdot m)/(\alpha \cdot S)}$ – sildķermeņa silšanas laika konstante, **s**.



3.6. att. **Žāvēšanas kameras ar elektrisku sildķermeni siltuma plūsmu sadalījums**

Žāvēšanas kamera ir statisks objekts ar pašizlīdzināšanās spēju. Tādēļ pie jebkuras konstantas sildītāja jaudas gaisa temperatūra kamerā izmainās aperiodiski pēc eksponenciāla likuma un vienmēr sasniedz noteiktu nostabilizējušos vērtību. Analītiski tas izpildās pie nosacījuma: $\mathbf{T_2 \geq 2T_1}$. Ja minētais nosacījums neizpildās, tad izvēlēti neatbilstoši kameras parametri vai kļūda vienādojuma (3.19) koeficientu aprēķinos.

Pielietojot Laplasa transformāciju diferenciālvienādojumam (3.19) pie nulles sākuma nosacījumiem, iegūstam žāvēšanas kameras **operatorvienādojumu** un **pārvades funkciju**:

$$T_1^2 \cdot \tau_g(s) \cdot s^2 + T_2 \cdot \tau_g(s) \cdot s + \tau_g(s) = K_{obj} \cdot P(s), \tag{3.20}$$

$$W_{obj}(s) = \frac{\tau_g(s)}{P(s)} = \frac{K_{obj}}{T_1^2 s^2 + T_2 s + 1}, \tag{3.21}$$

kur $\boldsymbol{\tau_g(s)}$ – kameras gaisa virstemperatūras attēla funkcija;

P(s) – sildelementa elektriskās jaudas attēla funkcija.

Apskatīsim temperatūras pārejas procesu modelēšanu mazgabarīta regulējamai žāvēšanas kamerai ar maināmu elektrisko spriegumu **U** kā ieejas iedarbi. Tā kā matemātiskajā modelī (3.19) ieejas lielums ir elektriskā jauda **P,** tad jāveic pārrēķins, izmantojot jaudas formulu: $\mathbf{P = U^2 / R}$, kur $\mathbf{R = U_{nom}^2 / P_{nom} = 220^2 / 500 = 97\ \Omega.}$ Pārrēķina realizāciju, izmantojot **Matlab** matemātiskās funkcijas, apskatīsim, sastādot modelēšanas blokshēmu.

Žāvēšanas kameras matemātiskais modelis ir pārvades funkcija (3.21). Aprēķinātais pārvades koeficients $\mathbf{K_{obj} = 0.32\ ^oC/W}$. Sildķermeņa aprēķinātā silšanas laika konstante **T = 10 min** un kameras gaisa aprēķinātā silšanas laika konstante $\mathbf{T_g = 4\ min.}$ Tad $\mathbf{T_1^2 = T \cdot T_g = 10{\cdot}4 = 40\ min^2}$ un $\mathbf{T_2 = T + T_g = 10 + 4 = 14\ min.}$

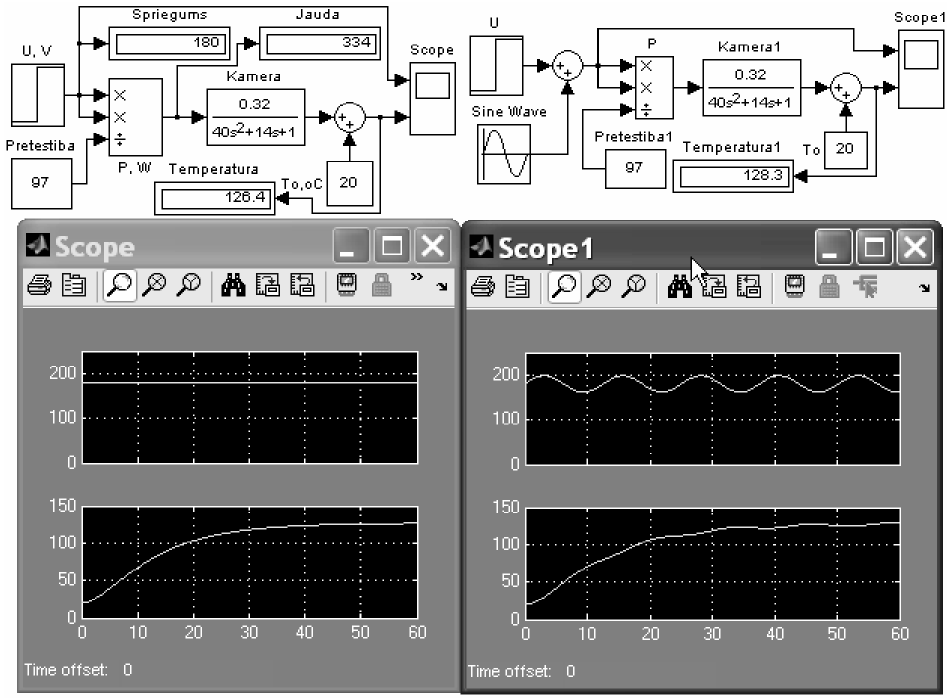
<u>Modelēšanas blokshēmas un raksturlīknes</u>. Lai iegūtu elektriskās žāvēšanas kameras pārejas procesa raksturlīknes, ja ieejas iedarbe ir konstants spriegums, sastādām modelēšanas blokshēmu (3.7.a att.). Izvēlamies **„Simulink”** bibliotēkas standarta blokus: **„Step**”, **„Constant”, „Display”, „Divide”, „Transfer Function”, „Sum”** un **„Scope”.** Bloku standarta nosaukumus var nomainīt. Ieklikšķinot uz bloka nosaukuma, piemēram, **„Step**”, to var nomainīt ar **„U, V”**, kas norāda, ka kāpņveida signāla ģenerators formē modeļa ieejas spriegumu. Līdzīgi var nomainīt pārējo bloku nosaukumus, jo tas padara blokshēmu uzskatāmāku un konkrētāku. Uzstādītie ciparu displeji **„Spriegums**”, **„Jauda”** un **„Temperatūra”** dod iespēju nolasīt modeļa attiecīgo ieejas un izejas lielumu nostabilizējušās skaitliskās vērtības.

Konfigurējam modeļa (3.7.a att.) blokus. Atveram kāpņveida signāla formēšanas bloku **„U, V”** un iestatām ieejas spriegumu **U = 180 V.** Lai aprēķinātu žāvēšanas kameras modelēšanas bloka **„Kamera”** ieejas lielumu – sildelementa elektrisko jaudu, izmantojam bloku **„P, W”,** kurš realizē ieejas lielumu reizināšanas un dalīšanas matemātiskās operācijas un realizē sekojošu algoritmu: **P = U x U ÷ R.** Divām reizināšanas ieejām pievadām spriegumu **U** un dalīšanas ieejai sildelementa elektrisko pretestību **R**, kuru formē konstanta signāla ģenerators **„Pretestība”**.

Atveram žāvēšanas kameras pārvades funkcijas bloku **„Kamera”,** kurā ievada pārvades koeficientu $\mathbf{K_{obj} = 0.32\ ^oC/W}$ un laika konstantes: $\mathbf{T_1^2 = 40\ min^2}$; $\mathbf{T_2 = 14}$ **min.** Atveram konstanta lieluma formēšanas bloku **„T_0”,** kurā iestata žāvēšanas kameras uzdoto gaisa sākuma temperatūru $\mathbf{\theta_0 = 20\ ^oC}$, kas pieņemta vienāda ar apkārtējās vides temperatūru**.**

Atveram logu **„Simulation parameters”** un izvēlamies simulācijas kopējo laiku $\mathbf{t_{sim} = 60}$ **min**, kas ir pietiekams, lai temperatūra kamerā pilnībā nostabilizētos.

a. **b.**



3.7. att. **Žāvēšanas kameras modelēšanas blokshēmas un raksturlīknes U = f(t) un θ_g = f(t), kur ieejas iedarbe ir elektriskais spriegums U(V) un izejas lielums – gaisa temperatūra θ_g (^{o}C) : a – konstantam spriegumam (U = 80 V); b – svārstīgam spriegumam (~U = 180 + 18·sin 0.5 t, V)**

Veicot simulāciju, uz osciloskopa **„Scope”** ekrāna iegūstam kameras ieejas lieluma – sprieguma **U** un izejas lieluma – gaisa temperatūras $\boldsymbol{\theta_g}$ pārejas procesa raksturlīknes: **U = f(t)** un $\boldsymbol{\theta_g}$ **= f(t)** (3.7.a att.). Displejs **„Jauda”** parāda kameras sildelementa elektrisko jaudu $\mathbf{P = U^2/R = 180^2/97 = 334\ W}$ un displejs **„Temperatūra”** parāda gaisa temperatūras $\boldsymbol{\theta_g}$ maksimālo nostabilizējušos skaitlisko vērtību:$\boldsymbol{\theta_{gs}=\theta_0+ +K_{obj}\cdot P=\theta_0 + +K_{obj}\cdot U^2/R = 20 + 0.32\cdot 180^2/97 = 126.4\ ^oC}$ (3.7.a att.).

Varam papētīt, kā sprieguma svārstības iespaido kameras gaisa silšanas procesu. Šai nolūkā modelēšanas blokshēmu papildināsim ar sinusoidāla signāla ģeneratoru **„Sine Wave”** un summatoru (3.7.b att.), kas dod iespēju formēt svārstīgu spriegumu ar brīvi izvēlētu svārstību frekvenci. Aktivizējamm bloku **„Sine Wave”.** Atveras bloka parametru logs, kurā ievadām sprieguma svārstību amplitūdu, piemēram, Δ**U = 18 V** un svārstību leņķisko frekvenci, piemēram, **ω = 0.5 rad/min.** Tā kā **ω = 2π·f,** tad sprieguma svārstību frekvence $\mathbf{f = \omega/2\pi = 0.5/2\cdot 3.14 = 0.08\ min^{-1}}$ un svārstību periods **T = 1/f = 1/0.08 = 12.5 min.** Svārstīgo spriegumu apraksta sekojoša funkcija: **~U = 180 + 18·sin 0.5 t.**

Veicot simulāciju, uz osciloskopa **„Scope1”** ekrāna iegūstam kameras ieejas lieluma – svārstīgā sprieguma ~**U** un izejas lieluma – gaisa temperatūras ~$\boldsymbol{\theta_g}$ pārejas procesa raksturlīknes: ~**U = f(t)** un **~**$\boldsymbol{\theta_g}$ **= f(t)** (3.7.b att.). Displejs **„Temperatūra1”** parāda gaisa temperatūras $\boldsymbol{\theta_g}$ nostabilizējušos skaitlisko vērtību simulācijas laika beigās: $\boldsymbol{\theta_{gs}}$**= 126.4** ^{o}**C** (3.7.b att.). Pārejas procesa raksturlīknes parāda, ka dotajam objektam ir laba pašizlīdzināšanās spēja, jo svārstoties spriegumam par ± **10 %**, temperatūras novirze salīdzinājumā ar nemainīgu spriegumu ir mazāka par **2%**.

3.5. Centrbēdzes regulatora kā svārstību posma modelēšana

Svārstību posms apraksta mehāniskas un elektriskas iekārtas, kurās rodas svārstīgs pārejas process ar rimstošu svārstību amplitūdu, piemēram, atsperes svārsts, kurā notiek enerģijas apmaiņa starp diviem elementiem – atsperi un atsvaru vai RLC svārstību kontūrs, kurā notiek enerģijas apmaiņa starp kondensatoru un spoli. Atsperes elastības enerģija pāriet atsvara kinētiskajā enerģijā un otrādi. Šīs apmaiņas rezultātā rodas rimstošs svārstību process. Līdzīgi uzlādēta kondensatora C elektriskā enerģija pāriet spoles L magnētiskajā enerģijā, kura savukārt pārlādē kondensatoru. Šajā enerģijas apmaiņas procesā rodas rimstošas elektriskās svārstības, kuru rimšanas ātrumu nosaka kontūra aktīvā pretestība R.

Svārstību posma darbības stabilitāti un pārejas procesa kvalitāti raksturo vairāki rādītāji, kā, piemēram, svārstīgums, svārstību rimšanas pakāpe, svārstību periods, maksimālais pārregulējums un pārejas procesa laiks. Svārstību posma vispārīgās īpašības un darbības rādītāji tika apskatīti automātikas pamatu kursā. Šeit galveno uzmanību pievērsīsim pārejas procesu modelēšanai Windows vidē.

Apskatīsim tipisku svārstību posmu – centrbēdzes rotācijas ātruma regulatoru, ko izmanto dīzeļmotora vārpstas rotācijas ātruma automātiskai stabilizācijai, mainoties mehāniskajai slodzei vai citiem faktoriem (3.8. att.).

Centrbēdzes regulatora ieejas lielums ir vārpstas rotācijas ātrums $\mathbf{\Omega_0} \pm \Delta\mathbf{\Omega}$ (rad/s), izejas lielums – degvielas vārsta pārvietojums $\mathbf{X_0} \pm \Delta$ **X (mm)**, kur $\mathbf{\Omega_0}$ un $\mathbf{X_0}$ – uzdotie lielumi, $\pm \Delta\mathbf{\Omega}$ un $\pm \Delta$ **X** – to izmaiņa regulēšanas procesā.

Lai samazinātu inerces spēku radītās svārstības, tiek lietots hidrauliskais slāpētājs, kurš sastāv no hidrocilindra **2** un regulējamas droseles **3**. Hidrocilindrs piepildīts ar eļļu, kas caur droseli var pārplūst no kāta telpas uz virzuļa telpu un otrādi. Eļļai pārplūstot no vienas virzuļa telpas uz otru rodas viskozā berze, kas slāpē inerces spēkus un samazina regulatora svārstīgumu. Samazinot droseles atvērumu, pieaug viskozā berze un palielinās svārstību slāpēšanas efektivitāte, taču vienlaicīgi samazinās regulatora ātrdarbība. Tādēļ, izvēloties svārstību slāpēšanas pakāpi, jāmeklē kompromiss starp regulatora darbības stabilitāti un ātrdarbību.

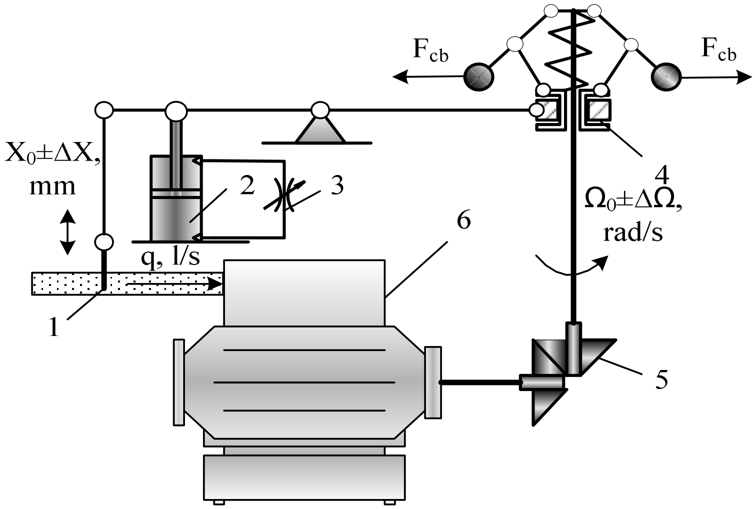
Centrbēdzes regulatora statiskā raksturlīkne $\mathbf{X_s = f(\Omega_s)}$ ir nelineāra, tādēļ apskatīsim tā darbību ierobežotā apgabalā ap uzdoto koordināti $\mathbf{X_0(\Omega_0)}$, sastādot regulatora **diferenciālvienādojumu mainīgo lielumu pieaugumiem** Δ **X** un $\Delta\mathbf{\Omega}$:

$$T_1^2 \frac{d^2\Delta X}{dt^2} + 2 \cdot \xi \cdot T_1 \frac{d\Delta X}{dt} + \Delta X = K_r \cdot \Delta\Omega, \qquad (3.22)$$

kur $\mathbf{K_r = \Delta X_s/\Delta\Omega_s}$ – centrbēdzes regulatora pārvades koeficients, **mm/(rad/s)**;

$\mathbf{T_1}$ – inerces spēku laika konstante, **s**;

ξ – svārstību rimšanas konstante.



3.8. att. **Centrbēdzes regulators dīzeļmotora rotācijas ātruma stabilizācijas sistēmā: 1- degvielas vārsts; 2 – svārstību slāpētājs; 3 – regulējama drosele; 4 – centrbēdzes regulators; 5 – pārvads; 6 – dīzeļmotors**

Pielietojot Laplasa transformāciju vienādojumam (3.22) pie nulles sākuma nosacījumiem, iegūstam **operatorvienādojumu**:

$$T_1^2 \cdot \Delta X(s) \cdot s^2 + 2\xi T_1 \cdot \Delta X(s) \cdot s + \Delta X(s) = K_r \cdot \Delta\Omega(s) \qquad (3.23)$$

un **pārvades funkciju**:

$$W(s) = \frac{\Delta X(s)}{\Delta\Omega(s)} = \frac{K_r}{T_1^2 \cdot s^2 + 2\xi T_1 \cdot s + 1}, \qquad (3.24)$$

kur Δ**X(s)** – degvielas vārsta atvēruma izmaiņas attēls;

Δ**Ω (s)** – rotācijas ātruma izmaiņas attēls;

s – Laplasa arguments.

Pielīdzinot regulatora pašoperatoru nullei, iegūstam tā **raksturīgo vienādojumu**:

$$D(s) = T_1^2 \cdot s^2 + 2\xi T_1 \cdot s + 1 = 0 \qquad (3.25)$$

Automātikas pamatu kursā noskaidrojām, ka svārstību posma raksturīgā vienādojuma saknes ir saistīti kompleksi skaitļi. Pēc raksturīgā vienādojuma saknēm varēsim noteikt centrbēdzes regulatora stabilitātes un darbības kvalitātes rādītājus.

Pārejas procesu modelēšanai izvēlamies centrbēdzes regulatora parametrus $\mathbf{T_1}$, $\mathbf{K_r}$, ξ. Aprēķinātā inerces spēku laika konstante $\mathbf{T_1 = 0.1\ s}$ un statiskais pārvades koeficients $\mathbf{K_r = -\ 0.7\ mm/(rad/s)}$. Svārstību rimšanas konstante ξ mainās atkarībā no droseles **3** atvēruma. Turpmākajā analīzē apskatīsim divus gadījumus: 1 – pie liela droseles atvēruma svārstību slāpēšanas efektivitāte ir maza un rimšanas konstante arī ir maza, piemēram, $\xi_1 = \mathbf{0.4}$; 2 – pieverot droseli, palielinām rimšanas konstanti līdz $\xi_2 = \mathbf{0.8}$, ko uzskatīsim par optimālu.

Koeficienta $\mathbf{K_r}$ negatīvā zīme norāda, ka regulatora izejas un ieejas lielumi mainās pretēji. Ja rotācijas ātrums samazinās, tad degvielas padeves vārsta atvērums palielinās un otrādi. Tā kā regulatora dinamikas analīzē svarīgaka ir tā izejas un ieejas lielumu izmaiņas savstarpējā sakarība, nevis izmaiņas virziens, tad turpmāk operēsim tikai ar koeficienta $\mathbf{K_r}$ absolūto vērtību. Tas padara vienkāršāku un saprotamāku centrbēdzes regulatora kā svārstību posma pārejas procesu analīzi.

3.5.1. Centrbēdzes regulatora reakcija uz lēcienveida iedarbi

Modelēšanas programmas un raksturlīknes. Centrbēdzes regulatora raksturīgā vienādojuma polu aprēķinam un pārejas procesa raksturlīkņu vizualizācijai sastādām analīzes programmas **Matlab** komandlogā divām svārstību rimšanas konstantēm $\xi_1 = \mathbf{0.4}$ un $\xi_2 = \mathbf{0.8}$ (3.9. att.). Ievietojot centrbēdes regulatora pārvades funkcijā (3.24) skaitliskos lielumus, iegūstam sekojošus **matemātiskos modeļus**:

$$W_1 = \frac{0.7}{0.1^2 \cdot s^2 + 2 \cdot 0.4 \cdot 0.1 \cdot s + 1} = \frac{0.7}{0.01 \cdot s^2 + 0.08 \cdot s + 1}; \qquad (3.26)$$

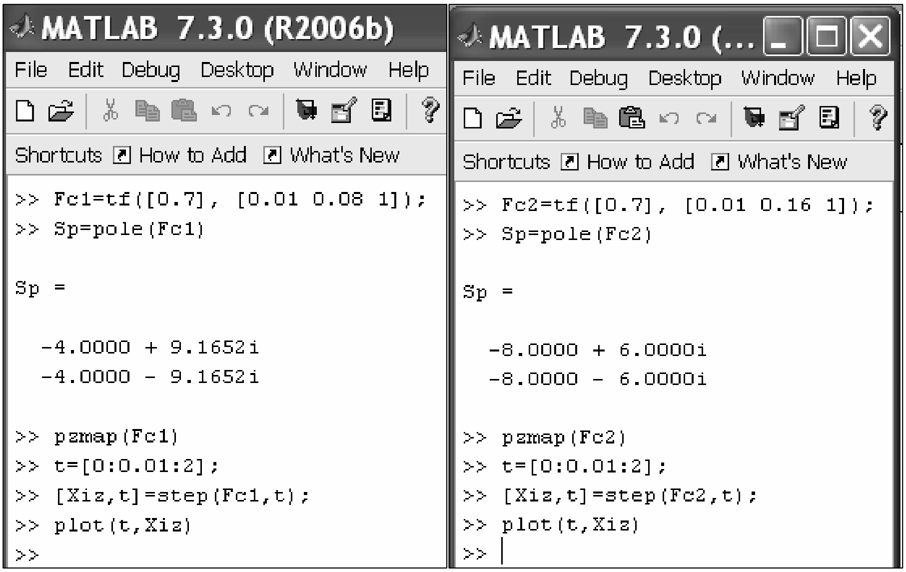
$$W_2 = \frac{0.7}{0.1^2 \cdot s^2 + 2 \cdot 0.8 \cdot 0.1 \cdot s + 1} = \frac{0.7}{0.01 \cdot s^2 + 0.16 \cdot s + 1}. \qquad (3.27)$$

Lai aprēķinātu centrbēdzes regulatora raksturīgā vienādojuma saknes (polus) un iegūtu to attēlojumu kompleksajā **s-plaknē**, atver **Matlab** komandlogu un sastāda programmu, izmantojot pārvades funkcijas un **Matlab** komandfunkcijas. Vispirms ievada pārvades funkcijas (3.26) iedarbes operatora **Q(s) = 0.7** un pašoperatora **D(s) = 0.01·s2 + 0.08·s + 1** parametrus sekojošā formā (3.9. att.):

>> Fc1= tf([0.7], [0.01 0.08 1]); →Enter,

>> Sp=pole(Fc1) →Enter.

a. **b.**



3.9. att. **Matlab komandlogā sastādītas programmas centrbēdzes regulatora polu un dinamisko raksturlīkņu aprēķinam un vizualizācijai pie dažādām svārstību rimšanas konstantēm ξ: a - ξ_1 = 0.4; b – ξ_2 = 0.8**

Ar **"tf"** tiek apzīmēta pārvades funkcija (transfer function). Visi programmas elementi jāizpilda precīzi ieskaitot iekavas, atstarpes un atdalīšanas zīmes. Ja ieviesusies kļūda, to var izlabot pirms „**Enter**". Pēc tam veikt labojumus nav iespējams. Tas nozīmē, ka ievadītā programmas daļa jānodzēš un jāievada no jauna.

Tālāk ievada komandfunkciju: **Sp = pole(Fc1)**, kas realizē pārvades funkcijas **Fc1** polu aprēķinu. Ar „**Enter**" komandlogā iegūstam divas kompleksi saistītas saknes (polus):

$$\mathbf{s_1 = \alpha_1 + \omega_1\, i = -4 + 9.17\, i} \text{ un } \mathbf{s_2 = \alpha_1 - \omega_1\, i = -4 - 9.17\, i}, \qquad (3.28)$$

kur $\boldsymbol{\alpha_1}$ – svārstību rimšanas koeficients, $\mathbf{s^{-1}}$;

$\boldsymbol{\omega_1}$ – svārstību leņķiskā frekvence, **rad/s**;

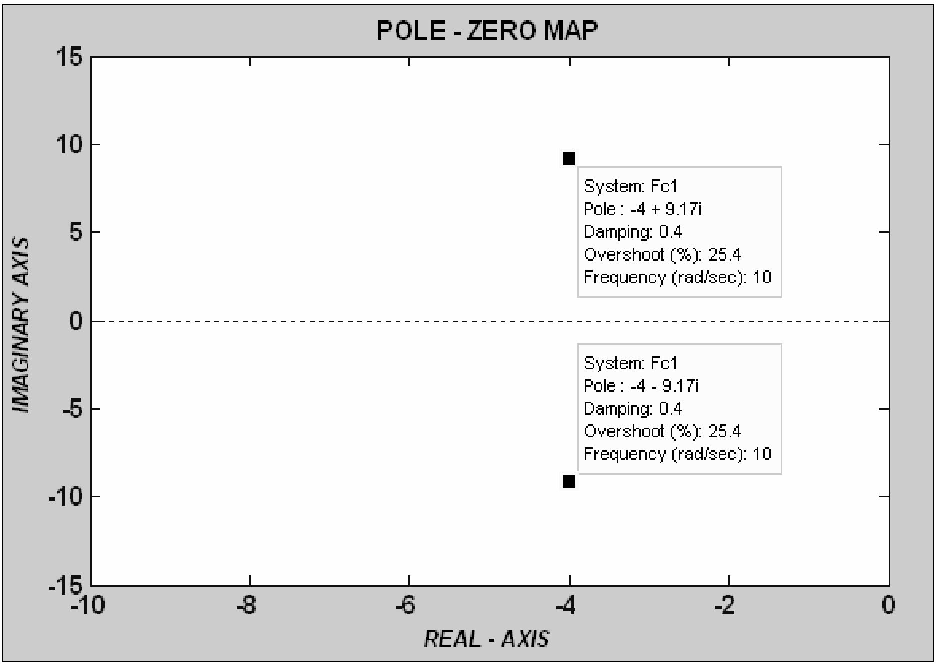
i – imaginārais skaitlis.

Lai iegūtu polu attēlojumu **s-plaknē**, ievadām komandfunciju: **pzmap(Fc1) → Enter**. Komandlogā parādās polu attēls (3.10. att.), kurš uzlabots un papildināts ar informāciju par polu parametriem un pārejas procesa rādītājiem, izmantojot papildus opcijas, piemēram, datu kursoru. Datu logā bez polu parametriem varam nolasīt svārstību rimšanas konstanti **(Damping)** – $\boldsymbol{\xi_1}$ **= 0.4** un maksimālo pārregulējumu **(Overshoot) – 25.4 %,** ko pieņemts apzīmēt ar $\boldsymbol{\sigma_{max}}$. Ar **(Frequency)** apzīmēts polu modulis $M_1 = \sqrt{\alpha_1^{\,2} + \omega_1^{\,2}} = \sqrt{(-4)^2 + 9.17^2} \approx 10$.

Varam izdarīt sekojošus secinājumus:

- pārejas process ir rimstošs un tātad stabils, jo poli izvietojas s-plaknes kreisajā pusplaknē un svārstību rimšanas koeficients $\boldsymbol{\alpha_1 = -4}$ ir negatīvs skaitlis;
- centrbēdzes regulatora pārejas process ir izteikti svārstīgs ar lielu pārregulējumu $\boldsymbol{\sigma_{max1}}$ **= 25.4 %,** kas pārsniedz maksimāli pieļaujamo – **20 %**.

Precīzāk centrbēdzes regulatora darbības stabilitāti un kvalitāti var novērtēt pēc tā izejas lieluma – degvielas vārsta pārvietojuma **Δ X (cm)** pārejas procesa raksturlīknes: $\mathbf{X_{iz} = \Delta X = f(t)}$**,** ja ieejas lielums – rotācijas ātrums **Ω** mainās lēcienveidīgi par lielumu **Δ Ω = 10 rad/s.**



3.10. att. **Centrbēdzes regulatora polu izvietojums kompleksajā s – plaknē, ja svārstību rimšanas konstante ξ_1 = 0.4**

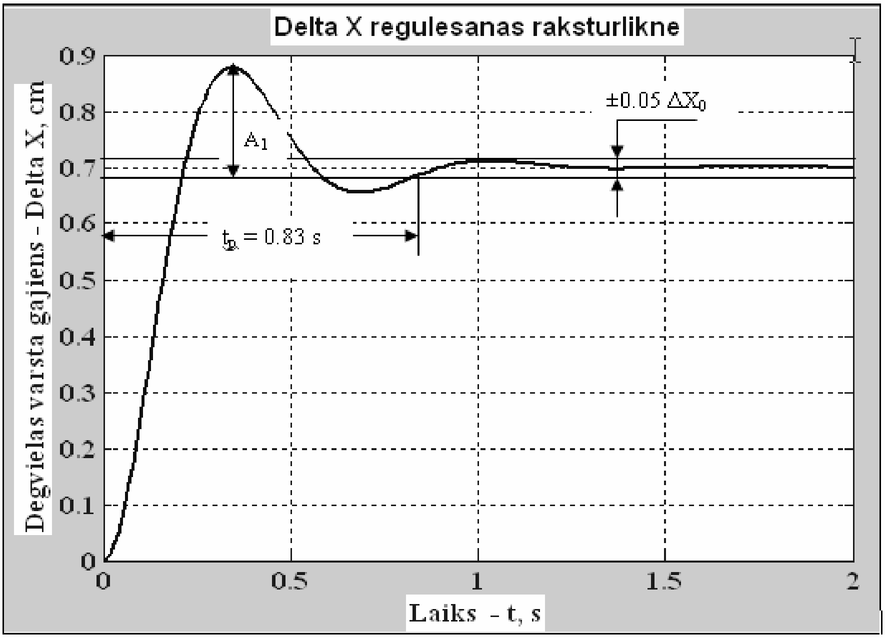
Pārejas procesa raksturlīknes aprēķinam un vizualizācijai Matlab komandlogā sastādām sekojošu programmu (3.9. att.):

>> t=[0:0.01:2]); →Enter

>> [Xiz,t]=step(Fc1,t); →Enter

>> plot(t, Xiz) →Enter

Pārejas procesa simulācijas laiks **t** ir no **0** līdz **2 s,** kas ir pietiekams, lai izejas lielums nostabilizētos. Punktu aprēķina solis ir **0.01 s**. Komandfunkcija **[Xiz, t]** realizē raksturlīknes punktu aprēķinu, bet - **plot(t, Xiz)** tās grafisko attēlojumu. **Matlab** komandlogā atveras pārejas procesa raksturlīknes logs (3.11. att.).



3.11. att. **Centrbēdzes regulatora degvielas vārsta atvēruma pārejas procesa raksturlīkne $\Delta X = f(t)$ pie lēcienveida rotācijas ātruma izmaiņas $\Delta\Omega$ = const, ja svārstību rimšanas konstante $\xi_1 = 0.4$**

Izmantojot dažādas papildus opcijas, iegūto attēlu varam uzlabot un papildināt, piemēram, uzstādot koordinātu tīklu, uzrakstus, fontus un krāsas, kā arī papildus līnijas un apzīmējumus. Attēlā parādīta regulējamā lieluma Δ **X** pirmā (maksimālā) pārregulējuma amplitūda $\mathbf{A_1}$ attiecībā pret nostabizējušos lielumu $\Delta \mathbf{X_0}$, kas dod iespēju aprēķināt maksimālo pārregulējumu: $\boldsymbol{\sigma_{max1} = A_1 / \Delta X_0 \cdot 100\ \% = 25.4\ \%}$.

3.11. attēlā iezīmēta **5 %** stabilizācijas zona: ± $\mathbf{0.05\ X_0}$, kas dod iespēju novērtēt pārejas procesa ātrdarbību, proti, izejas parametra pārejas procesa (regulēšanas) laiku: $\mathbf{t_p = 0.83\ s.}$ Raksturlīkne uzskatāmi parāda, ka pie mazas rimšanas konstantes $\boldsymbol{\xi_1 = 0.4}$**,** kad svārstību slāpēšanas drosele **3** (3.8. att.) ir gandrīz atvērtā stāvoklī un inerces spēki ir ievērojamā pārsvarā pār viskozās berzes radītajiem pretestības spēkiem, centrbēdzes regulatora pārejas process ir izteikti svārstīgs ar lielu maksimālo pārregulējumu un relatīvi lielu pārejas procesa laiku. Šādam regulatoram ir nepietiekama stabilitāte un ātrdarbība.

Lai uzlabotu centrbēdzes regulatora pārejas procesu, pievērsim droseli **3**, proti, palielināsim viskozo berzi, kas slāpē regulatora inerces spēkus. Pieņemsim, ka šīs darbības rezultātā svārstību rimšanas konstante palielinās no $\xi_1 = \mathbf{0.4}$ līdz $\xi_2 = \mathbf{0.8}$.

Līdzīgi kā iepriekšējā variantā **Matlab** komandlogā ievadām pārvades funkcijas (3.27) iedabes operatora **Q(s) = 0.7** un pašoperatora $\mathbf{D(s) = 0.01 \cdot s^2 + 0.16 \cdot s + 1}$ parametrus sekojošā formā (3.9. att.):

>> Fc2= tf([0.7], [0.01 0.16 1]); →Enter1

>> Sp=pole(Fc2) →Enter2

>> t=[0:0.01:2]); →Enter3

>>[Xiz,t]=step(Fc2,t); →Enter4

>> plot(t, Xiz) →Enter5

Pēc **Enter2** tiek veikts centrbēdzes regulatora polu aprēķins:

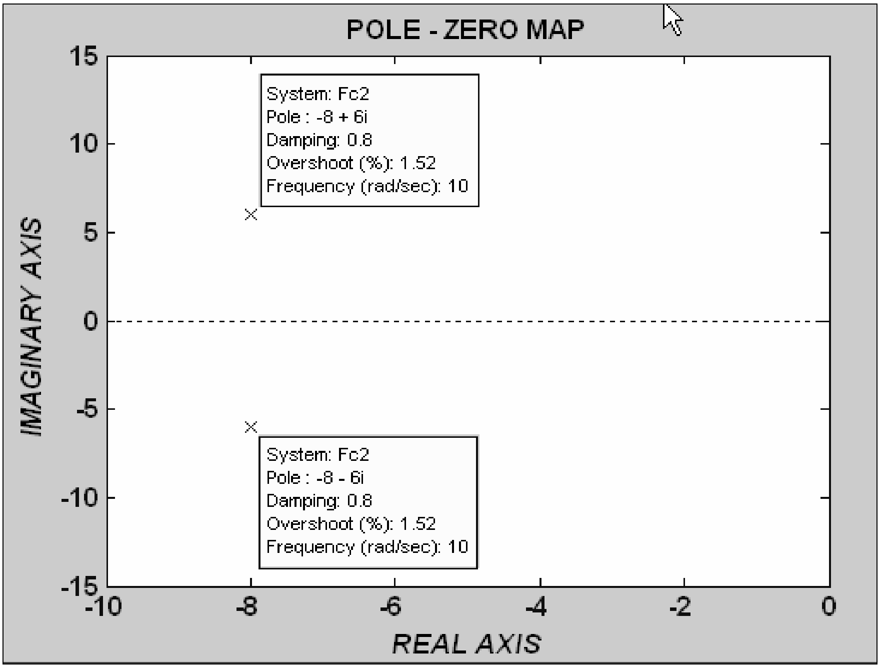
$$\mathbf{S_3 = \alpha_2 + \omega_2\, i = -8 + 6\, i} \text{ un } \mathbf{s_4 = \alpha_2 - \omega_2\, i = -8 - 6\, i}, \qquad (3.29)$$

bet pēc **Enter3** iegūstam polu grafisko attēlojumu kompleksajā **s-plaknē** (3.12. att.).

Salīdzinājumā ar iepriekšējo variantu svārstību rimšanas koeficients $\boldsymbol{\alpha_2}$ **palielinājies 2** reizes, bet svārstību leņķiskā frekvence $\boldsymbol{\omega_2}$ **samazinājusies** par **1/3.** Tas liecina, ka regulatora stabilitāte ir būtiski uzlabojusies. Poli ir ievērojami attālinājušies no stabilitātes robežas un pietuvojušies reālo skaitļu asij. Tā rezultātā ir palielinājusies centrbēdzes regulatora stabilitātes pakāpe un samazinājies pārejas procesa laiks.

Maksimālais pārregulējums $\boldsymbol{\sigma_{max2}} = \mathbf{1.52\ \%}$ samazinājies **25.4/1.52 = 16.7** reizes.

Centrbēdzes regulatora stabilitāti un darbības kvalitāti uzskatāmi atspoguļo pārejas procesa norise. Izejas lieluma pieauguma simulētā pārejas procesa raksturlīkne $\mathbf{\Delta X = f(t)}$ pie lēcienveida rotācijas ātruma izmaiņas $\mathbf{(t = 0,\ \Delta\Omega = 0;\ t > 0,\ \Delta\ \Omega = -10\ rad/s)}$ un svārstību rimšanas konstantes $\xi_2 = \mathbf{0.8}$ parādīta 3.13. attēlā. Redzam, ka palielinot svārstību rimšanas konstanti no $\xi_1 = \mathbf{0.4}$ līdz $\xi_2 = \mathbf{0.8}$ centrbēdzes regulatora darbības stabilitāte un kvalitāte ir ievērojami uzlabojusies.



3.12. att. **Centrbēdzes regulatora polu izvietojums kompleksajā s – plaknē, ja svārstību rimšanas konstante $\xi_2 = 0.8$**

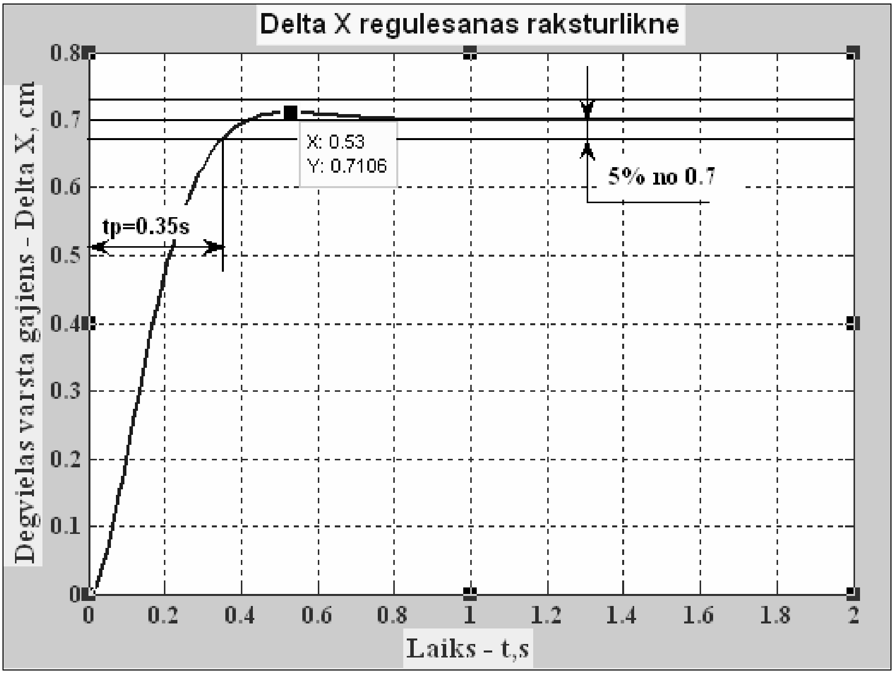
Noteiksim galvenos pārejas procesa kvalitāti raksturojošos rādītājus:

- maksimālais pārregulējums $\sigma_{\max 2} = A_1 / \Delta X_0 \cdot 100\% = 0.0106 / 0.7 \cdot 100\% = 1.51\%$, kas atbilst iepriekš noteiktajam (3.12. att.);
- pārejas procesa laiks $t_p \approx 0.35 s$ (3.13. att.).

Pārejas procesa laiku var aprēķināt arī izmantojot tuvinātu formulu:

$t_p \approx -(T_1 / \xi_2) \cdot \ln 0.05 = -0.1 / 0.8 \cdot \ln 0.05 = 0.36 s$. Laiks t_p tiek noteikts līdz momentam, kad regulējamais parametrs ΔX sasniedz 5 % stabilizācijas zonu.

Salīdzinājumā ar iepriekšējo variantu, centrbēdzes regulatora pārejas procesa laiks **t_{p2} = 0.35 s** samazinājies **0.83/0.35 = 2.4 reizes** un maksimālais pārregulējums ir mazāks par **5 %**. Pārejas process ir monotons ar mazu pārregulējumu. Šādu pārejas procesu var pieņemt par optimālu kā no stabilitātes, tā ātrdarbības viedokļa.

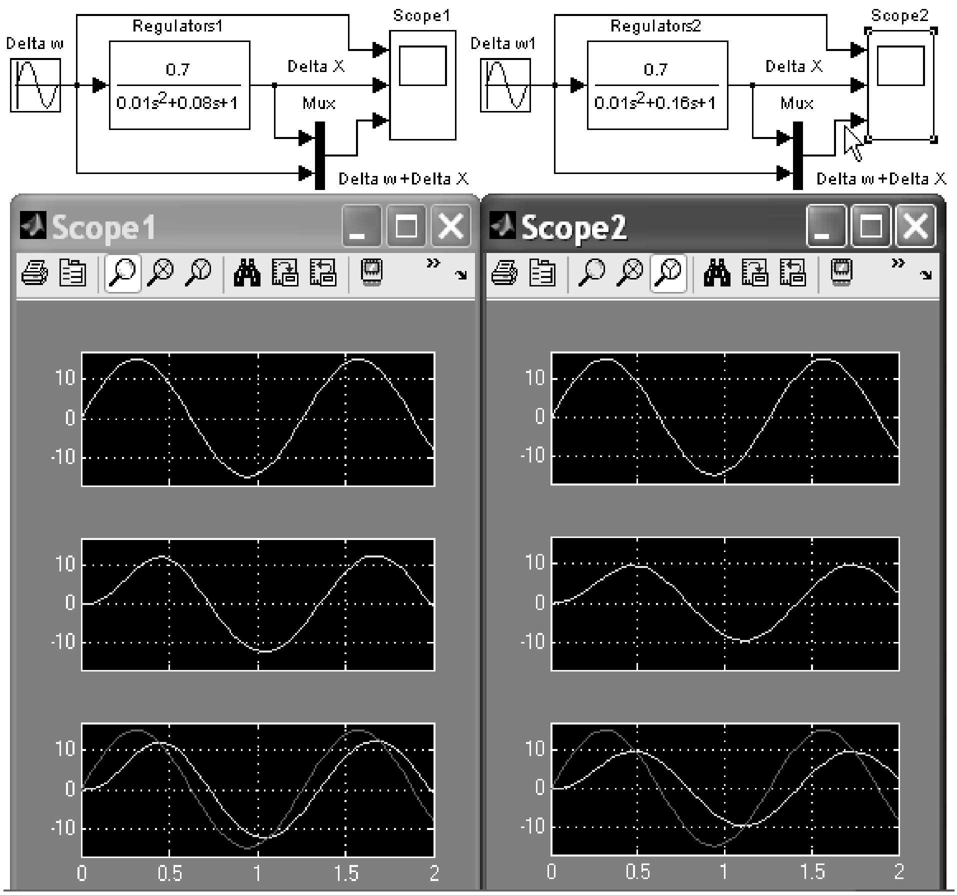


3.13. att. **Centrbēdzes regulatora degvielas vārsta atvēruma pārejas procesa raksturlīkne $\Delta X = f(t)$ pie lēcienveida rotācijas ātruma izmaiņas $\Delta\Omega$ = const., ja svārstību rimšanas konstante ξ_2 = 0.8**

3.5.2. Centrbēdzes regulatora reakcija uz harmonisku iedarbi

Darbojoties mainīgai perturbācijai – slodzei uz motora vārpstas, arī centrbēdzes regulatora pārejas process ir svārstīgs neatkarīgi no svārstību rimšanas konstantes ξ lieluma. Tādēļ svarīgi izpētīt centrbēdzes regulatora reakciju uz svārstīgu ieejas iedarbi. 3.14. attēlā parādītas regulatora modelēšanas blokshēmas un raksturlīknes, ja rotācijas ātrums svārstās sinusoidāli - **$\Delta\Omega = 14\cdot\sin 4t$** ap uzdotu līdzsvara stāvokli **Ω_0.**

a. **b.**



3.14. att. **Centrbēdzes regulatora frekvenču raksturlīknes $\Delta\Omega = A_{ie} \cdot \sin(\omega t)$ un $\Delta X = A_{iz} \cdot \sin(\omega t - \varphi(\omega))$, ja vārpstas rotācijas ātruma svārstību leņķiskā frekvence ω=4 rad/s un amplitūda A_{ie}=15 rad/s: a – svārstību rimšanas konstante ξ_1 = 0.4; b – svārstību rimšanas konstante ξ_2 = 0.8**

<u>Modelēšanas blokshēmas un raksturlīknes</u>. Lai iegūtu centrbēdzes regulatora pārejas procesa raksturlīknes, ja ieejas iedarbe ir sinusoidāli svārstīgs rotācijas ātrums, sastādām modelēšanas blokshēmas (3.14. att.). Izvēlamies **„Simulink”** bibliotēkas standarta blokus: **„Sine Wave**”, **„Transfer Function”**, **„Mux”** un **„Scope”.**

Sinusoidāla signāla ģenerators **„Sine Wave”** formē svārstīgu rotācijas ātruma izmaiņu **(Delta Ω)**. Bloki **„Transfer Function” (Regulators1)** modelē regulatora pārejas procesu, ja svārstību rimšanas konstante ξ_1 **= 0.4** un **(Regulators2)**, ja svārstību rimšanas konstante ξ_2 **= 0.8**. Signālu jaucēju jeb multipleksoru **„Mux”** izmanto, lai modeļa ieejas un izejas lielumus attēlotu vienā koordinātu sistēmā.

Konfigurējam modeļa (3.14.a att.) blokus. Atveram sinusoidāla signāla formēšanas bloku **„Delta Ω”.** Iestatām rotācijas ātruma svārstību amplitūdu – **15 rad/s** un svārstību leņķisko frekvenci – **4 rad/s.** Atveram bloku **„Regulators1”**, kurā ievada pārvades funkcijas **(3.26)** parametrus**.** Līdzīgi konfigurējam otra modeļa (3.14.b att.) blokus.

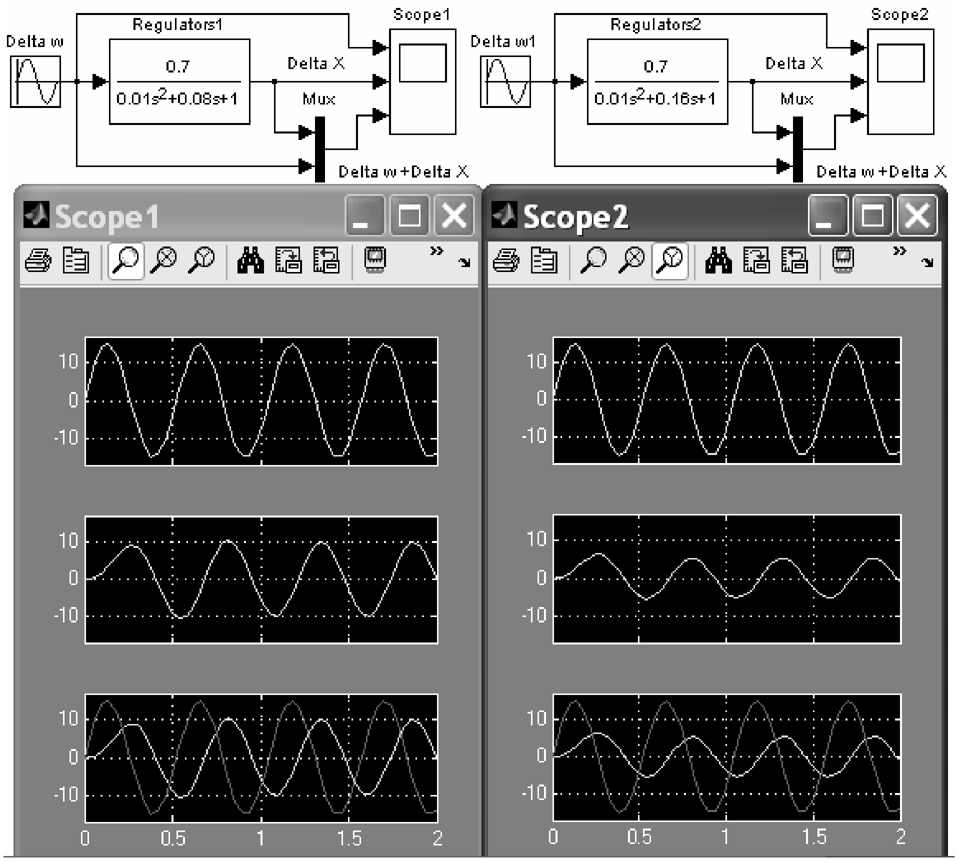
Atveram logu **„Simulation parameters”** un izvēlamies simulācijas kopējo laiku $\mathbf{t_{sim}}$ **= 2 s**, lai iegūtu vismaz vienu pilnu svārstību periodu.

Veicot simulāciju, uz osciloskopa **„Scope1”** ekrāna iegūstam centrbēdzes regulatora ieejas lieluma – rotācijas ātruma izmaiņas $\Delta\mathbf{\Omega}$ **= f(t)** un izejas lieluma – degvielas vārsta pārvietojuma $\Delta\mathbf{X}$ **= f(t)** pārejas procesa raksturlīknes (3.14.a att.). Pārejas procesa raksturlīknes parāda, ka pie relatīvi mazas svārstību rimšanas konstantes centrbēdzes regulatora izejas lieluma svārstību amplitūda ir proporcionāla ieejas lieluma svārstību amplitūdai un izejas lieluma izmaiņa aizkavējas attiecībā pret ieejas lieluma izmaiņu sakarā ar regulatora inerci.

Palielinot svārstību rimšanas konstanti no ξ_1 **= 0.4** līdz ξ_2 **= 0.8,** izejas lieluma svārstību amplitūda samazinās, bet tā aizkavēšanās attiecībā pret ieejas lieluma izmaiņu palielinās (3.14.b att.). Tātad notiek labāka svārstību slāpēšana.

Lai izpētītu ieejas lieluma leņķiskās svārstību frekvences $\boldsymbol{\omega}$ iespaidu uz izejas lieluma svārstību amplitūdu $\mathbf{A_{iz}(\omega)}$ un aizkavēšanās fāzi $\boldsymbol{\varphi(\omega)}$**,** simulējam pārejas procesu pie augstākas frekvences $\boldsymbol{\omega}$ **= 12 rad/s** (3.15. att.). Ja slāpēšanas konstante ir maza (ξ_1 **= 0.4),** ieejas lieluma amplitūda samazinās nebūtiski, bet aizkavēšanās fāze $\boldsymbol{\varphi(\omega)}$ pieaug ievērojami (3.15.a att.). Ja slāpēšanas konstante ir liela (ξ_2 **= 0.8),** ieejas lieluma amplitūda būtiski samazinās, bet aizkavēšanās fāze $\boldsymbol{\varphi(\omega)}$ vēl vairāk palielinās (3.15.a att.). Tātad centrbēdzes regulators labi slāpē augstāku frekvenču svārstības, ja izvēlēta atbilstoša svārstību slāpēšanas konstante.

a. b.



3.15. att. **Centrbēdzes regulatora frekvenču raksturlīknes $\Delta\Omega = 15 \cdot \sin(\omega\, t)$ un $\Delta X = A_{iz} \cdot \sin(12t - \varphi(\omega))$, ja vārpstas rotācijas ātruma svārstību leņķiskā frekvence $\omega = 12$ rad/s : a – svārstību rimšanas konstante $\xi_1 = 0.4$; b – svārstību rimšanas konstante $\xi_2 = 0.8$**

3.6. Ūdensapgādes iekārtas kā transportkavējuma posma modelēšana

Ja kādas tehnoloģiskas iekārtas ieejā padod lēcienveida iedarbi, uz kuru tā reaģē ar aizkavēšanos, tad saka, ka tā darbojas ar transportkavējumu. Šādas ierīces pārejas procesu apraksta transportkavējuma posms. Raksturīgi šādu posmu piemēri ir dažādu materiālu vai detaļu transportieri un konveijeri, pneimotransporta iekārtas, ūdens un siltuma apgādes trases, robottehnisko kompleksu transporta moduļi.

Tipisks transportkavējuma posms ir sūkņa iekārta ar garu ūdens pārneses cauruļvadu (3.16. att.). Ja cauruļvada garums no sūkņa līdz rezervuāram ir **l (m)** un ūdens plūsmas ātrums – **v (m/s)**, tad ūdens daudzuma pārneses transportkavējuma laiks no sūkņa ieslēgšanas momenta līdz ūdens ieplūdei rezervuārā ir **τ = l/v (s)**.

Apzīmējot sūkņa ražīgumu ar $\mathbf{Q_{ie}}$ $\mathbf{(m^3/s)}$ un cauruļvada šķērsgriezumu ar S $\mathbf{(m^2)}$, transportkavējuma laiku **τ (s)** var izteikt sekojoši:

$$\tau = \frac{l \cdot S}{Q_{ie}} . \tag{3.30}$$

Pieņemot, ka cauruļvadā plūsmas zudumu nav $\mathbf{(Q_{iz} = Q_{ie})}$, iegūstam cauruļvada **oriģinālvienādojumu**:

$$Q_{iz}(t) = Q_{ie}(t-\tau). \tag{3.31}$$

Pielietojot Laplasa transformāciju, no (3.31) iegūstam **operatorvienādojumu**:

$$Q_{iz}(s) = e^{-s\tau} Q_{ie}(s) \tag{3.32}$$

un **pārvades funkciju**:

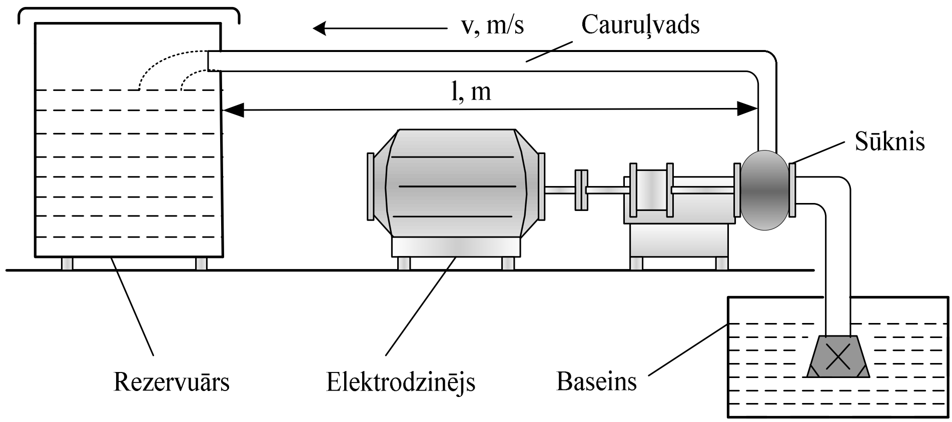
$$W(s) = \frac{Q_{iz}(s)}{Q_{ie}(s)} = e^{-s\tau}, \tag{3.33}$$

kur $\mathbf{Q_{iz}}$ **(s)** un $\mathbf{Q_{ie}}$ **(s) –** cauruļvada izejas un ieejas plūsmu attēli;

s -Laplasa arguments, $\mathbf{s^{-1}}$.

Automātikas pamatu kursā noskaidrojām, ka pārvades funkcija (3.33) tiešā veidā nepakļaujas analīzei, jo **s** ir komplekss skaitlis. Lai iegūtu pielietojamu matemātisko modeli, eksponenciālo funkciju izvirza Padē polinomālajā rindā. Padē rindas aproksimācijas kārtu izvēlas atkarībā no nepieciešamās precizitātes.

Lai pētītu transportkavējuma posma dinamiskās īpašības, izmanto **„Simulink”** bibliotēkas standarta bloku **„Transport Delay”,** kurā ievada transportkavējuma laiku **τ** un **Padē** aproksimācijas kārtu **k.** Parasti izvēlas **k = 4,** kas nodrošina pietiekamu precizitāti.



3.16. att. **Sūkņa iekārta ar ūdens pārneses transportkavējuma posmu**

Automātikas pamatu kursā noskaidrojām, ka padodot transporkavējuma posma ieejā lēcienveida iedarbi, arī izejas lielums izmainās lēcienveidīgi, bet ar aizkavēšanos laikā par transportkavējuma laiku **τ.** Izpētīsim cauruļvada, kā transportkavējuma posma reakciju uz lineāri un sinusoidāli mainīgu ieejas iedarbi.

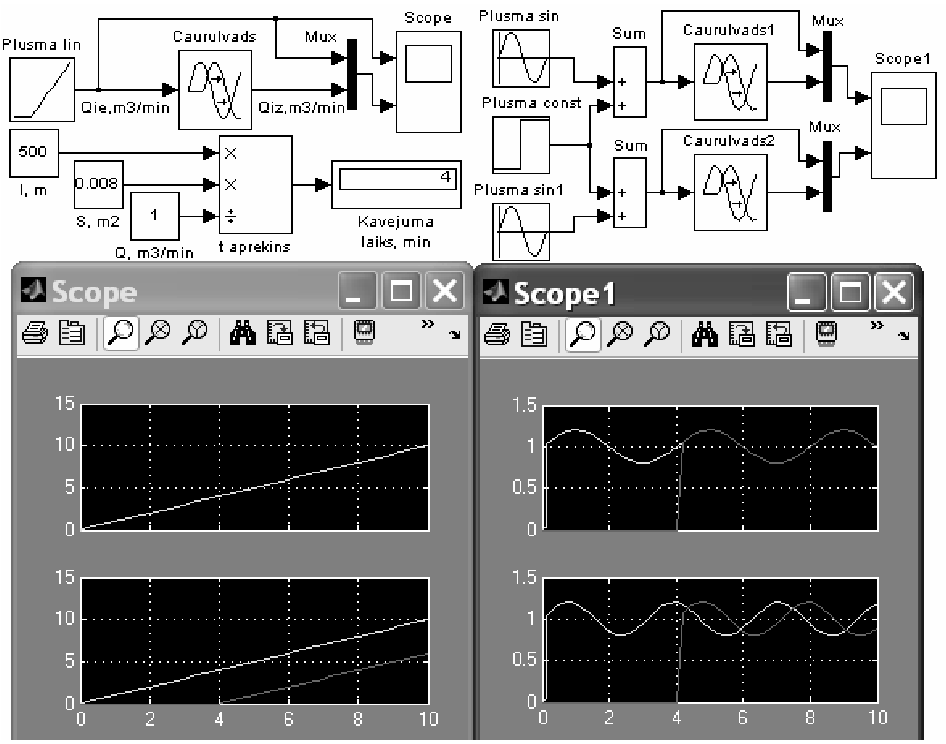
Modelēšanas blokshēmas un raksturlīknes. Cauruļvada kā transportkavējuma posma raksturlīkņu modelēšanai izvēlamies blokus no **„Simulink”** bibliotēkas: **„Constant”** - cauruļvada garuma **l,** šķērsgriezuma laukuma **S** un ūdens plūsmas **Q** ievadīšanai; **„Divide”** (**t aprēķins) -** transportkavējuma laika **τ** aprēķināšanai; ciparu displeju **„Display”** (**Kavējuma laiks**) – **τ** vizualizācijai; signālu avotus **„Linear”** – (**Plūsma lin**), **„Step”** (**Plūsma const**), **„Sine Wave”** (**Plūsma sin**) – lineāri mainīgas, konstantas un sinusoidāli mainīgas plūsmas formēšanai; **„Transport Delay”** (**Cauruļvads**) - cauruļvada pārejas procesa modelēšanai; **„Mux”** – divu raksturlīkņu attēlošanai vienā koordinātu sistēmā; **„Sum”** – signālu summēšanai; **„Scope”** – izejas lielumu oscilogrammu attēla iegūšanai (3.17. att.).

Konfigurējam blokus **l, S un Q,** ievadot tajos $\mathbf{l = 500\ m}$, $\mathbf{S = 8 \cdot 10^{-3}\ m^2}$ un $\mathbf{Q_{ie} = Q_{iz} = Q = 1\ m^3/min}$. Veicot simulāciju, iegūstam transportkavējuma laiku $\boldsymbol{\tau = (500 \cdot 8 \cdot 10^{-3})/1 = 4\ min.}$

Konfigurējam bloku **„Cauruļvads”**, ievadot $\boldsymbol{\tau = 4}$ **min** un Padē aproksimācijas kārtu **k = 4**. Blokā **„Plūsma lin”** ievadām $\mathbf{Q_{ie} = 1\ m^3/min}$.

Veicot simulāciju, iegūstam pārejas procesa raksturlīknes, kas parāda ūdens tilpuma **V (m^3)** pārneses dinamiku cauruļvadā (3.17.a att.). Cauruļvada izejā ūdens parādās pēc **4 min** no sūkņa ieslēgšanas brīža un katrā nākošajā laika momentā ūdens daudzums izejā ir par **4 m^3** mazāks salīdzinājumā ar pievadīto daudzumu.

a. **b.**



3.17. att. **Sūkņa iekārtas ar ūdens transportkavējuma laiku τ = 4 min pārejas procesa raksturlīknes: a –V_{ie}= f(t) un V_{iz}= f(t), ja Q_{ie}=1m^3/min= const.; b – Q_{iz} = f(t), ja Q_{ie1} = 1 + 0.2sin 1.57· t un Q_{ie2} = 1 + 0.2sin 2·t**

Izmantojot frekvenču regulējamo sūkņa elektropiedziņu var veidoties svārstīgs pārejas process. Izpētīsim svārstīgas plūsmas pārvades dinamiku pa cauruļvadu. Konstantai komponentei **Q_c = 1 m^3/min** uzklājam sinusoidālu komponenti ~ **Q = 0.2sin ωt,** kur **ω** – svārstību leņķiskā frekvence, **rad/min.** Modelēšanai izvēlamies divas **ω** vērtības.

1. Ieejas plūsmas svārstību periods **T = τ = 4 min.** Tad svārstību frekvence, kuru nosauksim par sinhrono frekvenci, $\mathbf{f_s = 1/T = 1/\tau = 0.25\ min^{-1}}$ un atbilstošā sinhronā leņķiskā frekvence, $\mathbf{\omega_s = 2\pi f_s = 2{\cdot}3.14{\cdot}0.25 = 1.57\ rad/min.}$

2. Leņķiskā frekvence $\mathbf{\omega > \omega_s = 2\ rad/min.}$

Modelēšanas blokshēma un raksturlīknes parādītas 3.17.b attēlā. Pie sinhronās frekvences $\mathbf{\omega = \omega_s = 1.57\ rad/min}$ cauruļvada izejas plūsmas svārstības sakrīt fāzē ar ieejas plūsmas svārstībām. Ja $\mathbf{\omega > \omega_s = 2\ rad/min,}$ izejas plūsmas svārstības aizkavējas attiecībā pret ieejas plūsmas svārstībām, ja $\mathbf{\omega < \omega_s}$ - apsteidz**.**

3.7. Integrējoša izpildmehānisma pārejas procesa modelēšana

Integrējošas ierīces var aprakstīt ar ideālu vai inerciālu integrējošu posmu, kura izvēli nosaka AVS pārējo komponentu inerce. Īpaši jāņem vērā vadības objekta inerce. Ja vadības objekta inerce ir par kārtu lielāka, nekā integrējošās ierīces inerce, tad tās modelēšanai var izvēlēties ideālu integrējošu posmu. Ja inerces ir salīdzināmas, tad jāizvēlas inerciālu integrējošu posmu.

Mazas inerces sistēmās, piemēram, ātrdarbīgās sekošanas sistēmās, pārejas procesa ilgums ir tikai dažas desmitdaļas sekundes. Izmantojot integrējošu elektrisku izpildmehānismu mazas inerces sistēmā, jāņem vērā tā elektromehāniskā inerce. Padodot šāda izpildmehānisma ieejā lēcienveida iedarbi, piemēram, ieslēdzot spriegumu, vārpstas rotācijas ātrums pieaug pakāpeniski pēc eksponenciāla likuma un tikai pēc zināma laika sasniedz maksimālo nostabilizējušos vērtību. Šīs aizkavēšanās inerci raksturo laika konstante **T, s**. Mazjaudas izpildmehānismiem un integrējošiem elektroniskajiem pastiprinātājiem **T = 0.02 – 0.1 s**. Taču arī šāda relatīvi maza inerce būtiski iespaido dinamiskos pārejas procesus ātrdarbīgās sistēmās.

Apskatīsim izpildiekārtu, kas sastāv no motorreduktora izpildmehānisma un vārsta, ar kuru automātiski regulē degvielas padevi uz koģenerācijas iekārtas iekšdedzes motoru, lai stabilizētu tā izejas vārpstas rotācijas ātrumu (3.18. att.).

Motorreduktora izpildmehānisms (turpmāk izpildmehānisms) sastāv no divfāžu asinhronā elektrodzinēja un zobratu reduktora. Ieejas lielums ir regulējams maiņspriegums **U (V)**, kuru padod uz vadības tinumu. Ierosmes tinumam pieslēdz konstantu maiņspriegumu **U_i (V)**. Izejas lielums ir reduktora vārpstas pagrieziena leņķis **φ (rad),** kas nosaka vārsta stāvokli un degvielas padeves daudzumu **q (l/s)**.

Dinamisko sakarību starp izejas lielumu **φ** un ieejas lielumu **U** apraksta pārvades funkcija:

$$W(s) = \frac{\varphi(s)}{U(s)} = \frac{k_{im}}{T_{im} \cdot s^2 + s}, \tag{3.34}$$

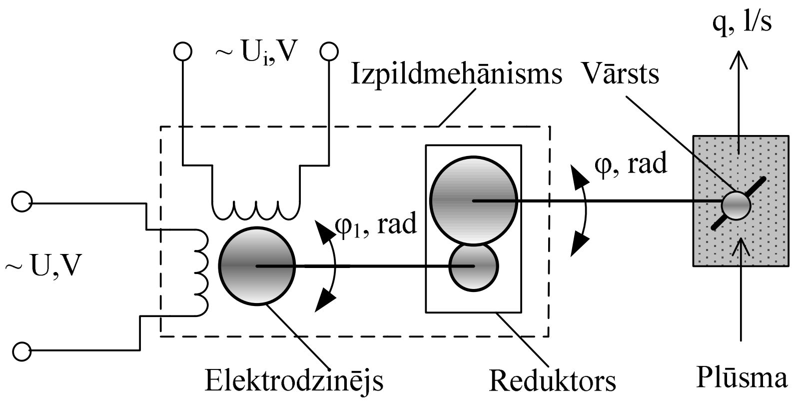
kur **φ(s)** – izejas vārpstas pagrieziena leņķa attēls;

U(s) – vadības sprieguma attēls;

s – Laplasa arguments, $\mathbf{s^{-1}}$;

$\mathbf{k_{im} = \omega/U}$ – izpildmehānisma ātruma koeficients, **(rad/s)/V**;

$\mathbf{T_{im}}$ – izpildmehānisma elektromehāniskā laika konstante, **s**.



3.18. att. **Izpildmehānisms ar divfāzu asinhrono elektrodzinēju un reduktoru**

Elektromehānisko laika konstanti galvenokārt nosaka rotējošo masu inerces moments, kas reducēts pie elektrodzinēja vārpstas **J_r (kg·m²)**, nominālais rotācijas ātrums **ω_{1nom} (rad/s)** un nominālā jauda **P_{1nom} (W)**. Zinot minētos parametrus, var aptuveni novērtēt elektromehānisko inerci: $\mathbf{T_{im} \approx (J_r \cdot \omega_{1nom}^2)/P_{1nom}}$.

Ja vadības objekta laika konstante daudzkārt lielāka par izpildmehānisma laika konstanti (**T_{obj} » T_{im}**), tad pēdējo var neņemt vērā. Mazinerciāliem vadības objektiem, kā gāzmotoram, kura laika konstante ir salīdzināma ar izpildmehānisma laika konstanti, pēdējo nedrīkst ignorēt. jo var rasties būtiskas kļūdas koģenerācijas iekārtas vadības sistēmas pārejas procesu modelēšanā.

<u>Modelēšanas blokshēmas un raksturlīknes</u>. No kataloga izvēlamies mazjaudas izpildmehānismu DKIR – 1 ar sekojošiem parametriem:

- mehāniskā jauda uz elektrodzinēja vārpstas **P_{1nom} = 1 W**;
- nominālais vadības spriegums **U_{nom} = 24 V**;
- ierosmes spriegums **U_i = 220 V**;
- elektrodzinēja vārpstas sinhronais rotācijas ātrums **ω_{1s} = 157 rad/s**;
- reduktora izejas vārpstas rotācijas ātrums **ω_{nom} = 1.57 rad/s**;
- elektromehāniskā laika konstante **T_{im} = 0.2 s**;
- ātruma koeficients **$k_{im} = \omega_{nom} / U_{nom}$ = 1.57/24 = 0.065 (rad/s)/V.**

Ievietojot dotos un aprēķinātos lielumus **T_{im}** un **k_{im}** pārvades funkcijā (3.34), sastādām pārejas procesu modelēšanas blokshēmas lēcienveida un sinusoidālam ieejas signālam (3.19. att., 3.21. att.).

No **„Simulink”** bibliotēkas izvēlamies blokus: **„Step” (±U); „Sine Wave” (~Usin); „Transfer Function” (Izpildmehānisms); „Saturation” (+U** un **-U); „Switch” (Galaslēdzis1** un **Galaslēdzis2), „Sum”, „Mux”** un **„Scope”.**

Ar signāla ierobežotājiem **„Saturation”** nodala vadības sprieguma **U** pozitīvo „**+U**” un negatīvo „**-U**” komponenti, jo „**Galaslēdzis1**” tiek vadīts ar izpildmehānisma izejas vārpstas pagrieziena leņķa **φ** pozitīvo komponenti „+**φ**” , bet **„Galaslēdzis2” –** ar negatīvo komponenti **„-φ”.** Galaslēdži izslēdz izpildmehānisma elektrodzinēju, kad tā izejas vārpsta pagriežas par maksimālo leņķi attiecībā pret vidusstāvokli: +**φ_{max}** - vārsts pilnīgi atvērts: - **φ_{max}** - vārsts pilnīgi aizvērts.

Konfigurējam modelēšanas blokshēmas (3.19. att.) blokus. Blokā „±U” ievadām laiku **t** un vadības spriegumu **U (0 < t < 4 s, U = +20 V; t ≥ 4 s, U = -20 V)**. Ja vadības spriegums ir pozitīvs, tad tas sakrīt fāzē ar ierosmes spriegumu, ja negatīvs, tad tas ir pretfāzē un izpildmehānisma vārpsta griežas pretējā virzienā.

Atveram sprieguma ierobežotāju **„+U”** un iestatām pozitīvā sprieguma apakšējo robežu: **„0”** un augšējo robežu: **„25”.** Blokā **„-U”** ievadām negatīvā sprieguma apakšējo robežu **„-25”** un augšējo robežu **„0”.**

Piezīme. Jāievēro, ka negatīva skaitļa vērtība ir zemāka, ja lielāka ir tā absolūtā vērtība, piemēram, **-5 > -25.**

Galaslēdžus var modelēt ar bloku **„Switch”- slēdzis, pārslēdzējs**, kuram ir trīs ieejas un viena izeja. Vidējai ieejai pievada vadības signālu, kurš automātiski pārkomutē 1. un 3. ieeju, savienojot vai atslēdzot no izejas. Lai konfigurētu galaslēdžus, izvēlas izpildmehānisma vārpstas maksimālo pagrieziena leņķi attiecībā pret vidusstāvokli pozitīvā virzienā **$+\varphi_{max}$ = 1 rad** (vārsts atvērts) un negatīvā virzienā **$-\varphi_{max}$ = -1 rad** (vārsts aizvērts). Vārsta pilns darba gājiens **φ_g = 2 rad.**

Aktivizējam bloku „**Galaslēdzis1”.** Atveras logs – **Block Parameters: Switch.** Izvēlamies kritērijus 1. ķēdes saslēgšanai – **Criteria for passing first imput.** Tiek piedāvāti 3 kritēriji:

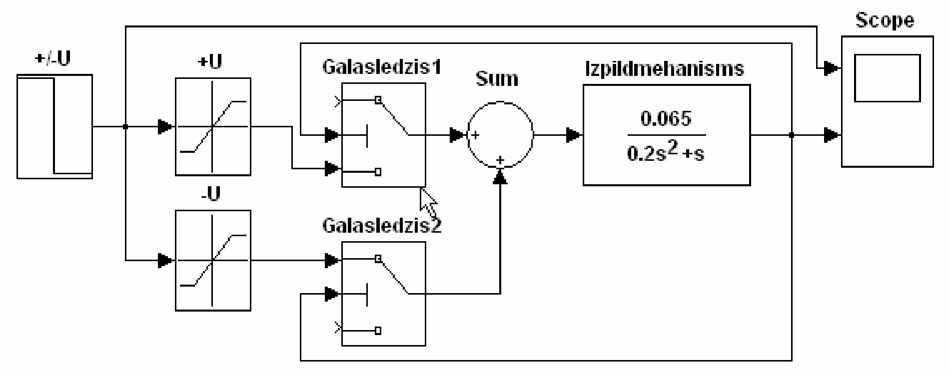
- **u2> =** - vadības signāls lielāks vai vienāds ar izvēlēto nostrādes slieksni **u_s (threshold);**
- **u2>** - vadības signāls lielāks par izvēlēto nostrādes slieksni **u_s;**
- **u2~=** - vadības signāls nav vienāds ar nostrādes slieksni **u_s.**

Dotajā gadījumā kā piemērotāko izvēlamies pirmo kritēriju: 1. ķēde saslēdzas, ja izpildmehānisma izejas vārpstas pagrieziena leņķis pozitīvā virzienā kļūst vienāds vai lielāks par **1 rad.** Dotajā gadījumā, pagriežoties vārpstai par leņķi **$+\varphi_{max}$ = 1 rad,** izpildmehānismam ir jāizslēdzas, tādēļ izvēlamies 3. ķēdi, kura darbojas pretēji 1. ķēdei. Ja 1. ķēde ir pārtraukta, tad 3. ķēde ir saslēgta. Saslēdzoties 1. ķēdei, 3. ķēde pārtraucas.

Ailē **Threshold** ievadām nostrādes slieksni. Lai vārpstas pagrieziena leņķis **$+\varphi_{max}$** nepārsniegtu **1 radiānu**, izvēlamies nostrādes slieksni **u_s = 0.8 rad.** Šādu izvēli nosaka izpildmehānisma izskreja, kuru rada tā elektromehāniskā inerce. Pēc sprieguma izslēgšanas vārpsta vēl pagriežas par leņķi, kas rada pozicionēšanas kļūdu un dotajam izpildmehānismam ir aptuveni **0.2 rad.** Vadības signāls, kurš kontrolē vārpstas pagrieziena leņķi, tiek padots no izpildmehānisma izejas caur atgriezenisko saiti.

Līdzīgi tiek konfigurēts **„Galaslēdzis 2”.** Atšķirība tāda, ka šis galaslēdzis kontrolē izpildmehānisma vārpstas maksimālo pagrieziena leņķi negatīvā (pretējā) virzienā: **-φ_{max}.** Tādēļ ailē **Threshold** jāievada nostrādes slieksni: **u_s = - 0.8 rad.** un 3. ķēdes vietā jāizmanto 1. ķēde, jo slēdzis uz negatīvu vadības signālu reaģē otrādi.

Blokā **„Izpildmehānisms”** ievadām pārvades funkcijas (3.34) parametrus.

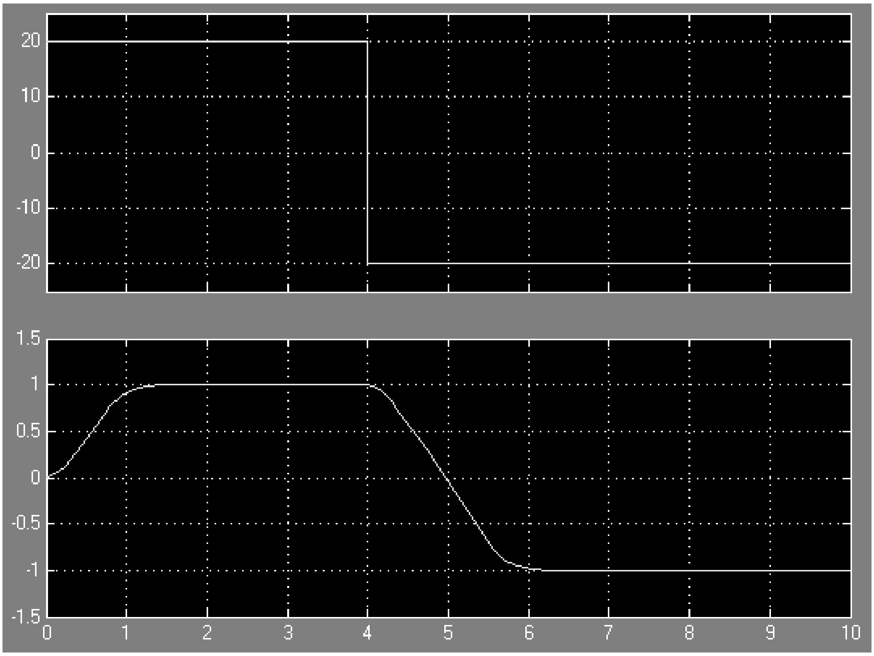


3.19. att. **Elektriska izpildmehānisma ar galaslēdžiem modelēšanas blokshēma, ja vadības spriegums mainās lēcienveidīgi (U = ±20V)**

Veicot simulāciju, uz osciloskopa **„Scope”** ekrāna iegūstam izpildmehānisma ieejas un izejas lielumu pārejas procesa raksturlīknes, ja vadības spriegums mainās lēcienveidīgi (3.20. att.). Laika intervālā **t = (0 – 4) s** spriegums **U = +20 V**. Laika momentā **t = 4 s** notiek sprieguma zīmes maiņa no **U = +20 V** uz **U = -20 V,** kā rezultātā izpildmehānisms maina griešanās virzienu. Dotajā gadījuma ar zīmes maiņu jāsaprot vadības sprieguma **U** fāzes maiņu attiecībā pret ierosmes spriegumu **U_i.** Ja vadības sprieguma fāze sakrīt ar ierosmes sprieguma fāzi, tad vārpsta griežas nosacīti pozitīvā virzienā, ja spriegumi ir pretfāzē, tad vārpsta griežas pretēji.

Šāda lēcienveida vadības sprieguma maiņa raksturīga divpozīciju regulēšanas sistēmās, kurās vadības objekta izejas lielumam tiek kontrolēti divi līmeņi – augšējais un apakšējais.

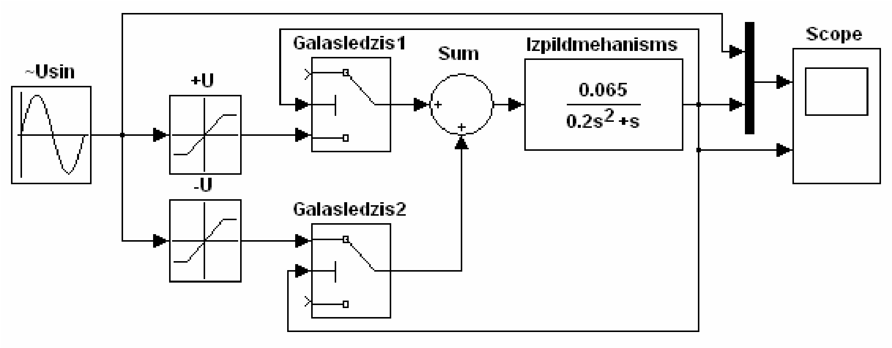
Kā piemēru var apskatīt rezervuāru, kurš nodrošina nepieciešamās ūdens rezerves siltumnīcai. Sistēmas palaišanas brīdī rezervuārā līmenis ir zem **H_{min}** Izpildmehānisms ātri atver ūdens padeves vārstu. Sasniedzot pilnu atvērumu (**t > 1 s**), nostrādā galaslēdzis un izpildmehānismu izslēdz. Caur pilnīgi atvērto vārstu rezervuārā intensīvi ieplūst ūdens un līmenis paaugstinās. Kad tas sasniedz maksimālo **H_{max} (t = 4 s),** vadības spriegums maina polaritāti, kā rezultātā izpildmehānisms ieslēdzas pretējā virzienā un aizver ūdens padeves vārstu. Galējā stāvoklī izpildmehānismu izslēdz otrs galaslēdzis (3.20. att.).



3.20. att. **Elektriska izpildmehānisma ar galaslēdžiem vadības sprieguma U(t) un vārpstas pagrieziena leņķa φ(t) pārejas procesa raksturlīknes, ja vadības spriegums mainās lēcienveidīgi (U = ±20V)**

Nepārtrauktās regulēšanas sistēmās , piemēram, koģenerācijas iekārtas vadības sistēmā, degvielas padeves vārsta izpildmehānisma ieejā rodas svārstīgs signāls. Noskaidrosim inerciāla integrējoša izpildmehānisma reakciju uz sinusoidāli mainīgu ieejas spriegumu **$U = U_{max} \cdot \sin \omega t$**.

Modelēšanas blokshēmā nomainām lēcienveida signāla avotu ar sinusoidāla signāla ģeneratoru **„~U sin”** (3.21. att.). Konfigurējam ģeneratoru, ievadot tajā vadības sprieguma svārstību amplitūdu U_{max} **= 20 V** un svārstību leņķisko frekvenci **ω = 1 rad/s.** Tad ģenerators formē sinusoidālu spriegumu **U = 20·sin 1t.**

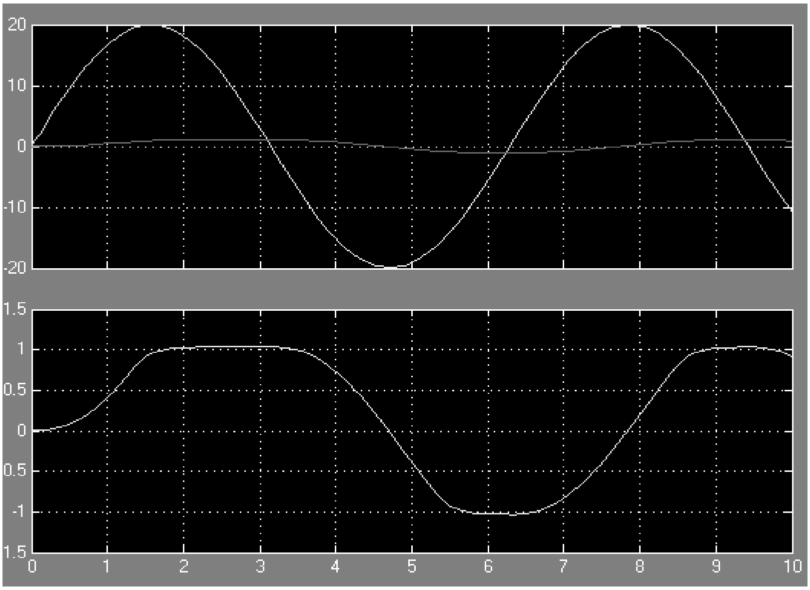


3.21. att. **Elektriska izpildmehānisma ar galaslēdžiem modelēšanas blokshēma, ja vadības spriegums mainās sinusoidāli (U =** U_{max}**·sin ωt)**

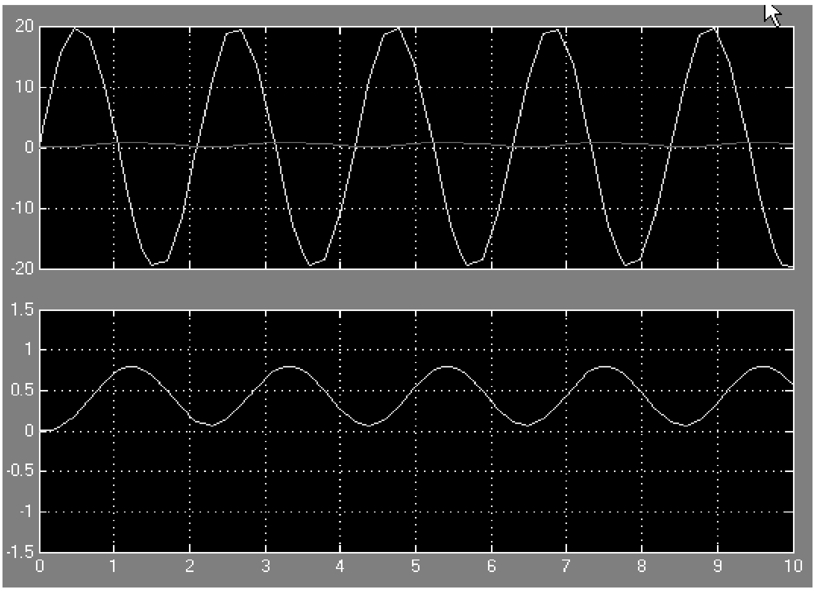
Veicot simulāciju, uz osciloskopa **„Scope"** ekrāna iegūstam izpildmehānisma vadības sprieguma un izejas vārpstas pagrieziena leņķa dinamiskās raksturlīknes (3.22. att.). Augšējā logā redzama vadības sprieguma izmaiņas raksturlīkne **U = 20·sin·1t** un vārpstas pagrieziena leņķa izmaiņas raksturlīkne **φ = f(t)** vienādos mērogos. Apakšējā logā parādīta raksturlīkne **φ = f(t)** palielinātā mērogā.

Izpildmehānisms darbojas pašsvārstību režīmā ar svārstību amplitūdu **±1 rad.** Vārsts periodiski tiek atvērts un aizvērts. Šāds režīms ir nevēlams. Tā cēlonis var būt neatbilstība starp izpildmehānisma un vadības objekta inerci (izpildmehānisms ir pārāk ātrdarbīgs) vai uz objektu darbojas iepriekš neparedzētas perturbācijas.

Izpildmehānisma reakcija uz vadības spriegumu ar 3rezes lielāku svārstību frekvenci **U = 20·sin·1t** parādīta 3.23. attēlā. Vārpstas svārstību amplitūda ir būtiski samazinājusies. Izpildmehānisms labi slāpē augstāku frekvenču svārstības.



3.22. att. **Elektriska izpildmehānisma ar galaslēdžiem vadības sprieguma U(t) un vārpstas pagrieziena leņķa φ(t) pārejas procesa raksturlīknes, ja vadības spriegums mainās sinusoidāli (U = 20·sin 1t)**



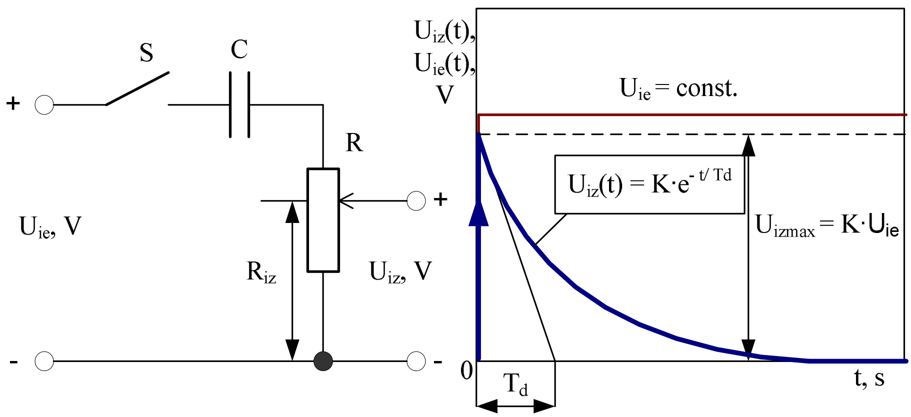
3.23. att. **Elektriska izpildmehānisma ar galaslēdžiem vadības sprieguma U(t) un vārpstas pagrieziena leņķa φ(t) pārejas procesa raksturlīknes, ja vadības spriegums mainās sinusoidāli (U = 20·sin 3t)**

3.8. Diferencējoša CR filtra pārejas procesu modelēšana

Diferencējošs posms reaģē uz mainīgu ieejas signālu. Ja ieejas signāls $\mathbf{X_{ie} = const.}$, tad izejas signāls $\mathbf{X_{iz} = 0}$. Diferencējoša posma izejas signāls ir proporcionāls ieejas signāla izmaiņas ātrumam. Ja reāla diferencējoša posma ieejā padod lēcienveida iedarbi, tad izejā iegūst impulsu, kurš pakāpeniski norimst.

Tipiski diferencējoša posma piemēri: **CR** elektriskā ķēde, kuras ieejas elements ir kondensators ar kapacitāti **C**, bet izejas elements – rezistors ar pretestību **R**; diferencējošais transformators; diferencējošais žiroskops; tahoģenerators, ja tā ieejas lielums ir vārpstas pagrieziena leņķis.

Kā piemēru apskatīsim vienkāršotu diferencējoša **CR** filtra shēmu (3.24. att.), ko izmanto kā korekcijas ķēdi AVS stabilitātes un darbības kvalitātes uzlabošanai.



3.24. att. **Diferencējoša CR filtra elektriskā shēma un izejas sprieguma $U_{iz}(t)$ izmaiņas raksturlīknes, ja ieejas spriegums $U_{ie}(t)$ mainās lēcienveidīgi**

Ieslēdzot slēdzi **S**, filtra ieejā lēcienveidīgi tiek padots līdzstrāvas spriegums**(**$\mathbf{t<0,\ U_{ie} = 0;\ t \geq 0,\ U_{ie} = const.}$**)**. Tā izejā iegūst sprieguma impulsu ar amplitūdu $\mathbf{U_{iz\,max} \leq U_{ie}}$, kurš pakāpeniski norimst līdz nullei pēc eksponenciāla likuma.

Pārejas procesu filtra ķēdē apraksta **pirmās kārtas diferenciālvienādojums** ar konstantiem koeficientiem:

$$T_d \cdot \frac{dU_{iz}(t)}{dt} + U_{iz}(t) = K \cdot T_d \cdot \frac{dU_{ie}(t)}{dt}, \qquad (3.35)$$

kur **K = U_{izmax} / U_{ie} = R_{iz} / R** - filtra pārvades koeficients;

T_d = R C - diferencēšanas laika konstante, **s.**

Pielietojot Laplasa transformāciju vienādojumam (3.35) pie nulles sākuma nosacījumiem, iegūstam **operatorvienādojumu**:

$$T_d \cdot U_{iz}(s) \cdot s + U_{iz}(s) = K \cdot T_d \cdot s \cdot U_{ie}(s) \qquad (3.36)$$

un **pārvades funkciju**:

$$W(s) = \frac{U_{iz}(s)}{U_{ie}(s)} = \frac{K \cdot T_d \cdot s}{T_d \cdot s + 1}. \qquad (3.37)$$

<u>Modelēšanas blokshēmas un raksturlīknes.</u>

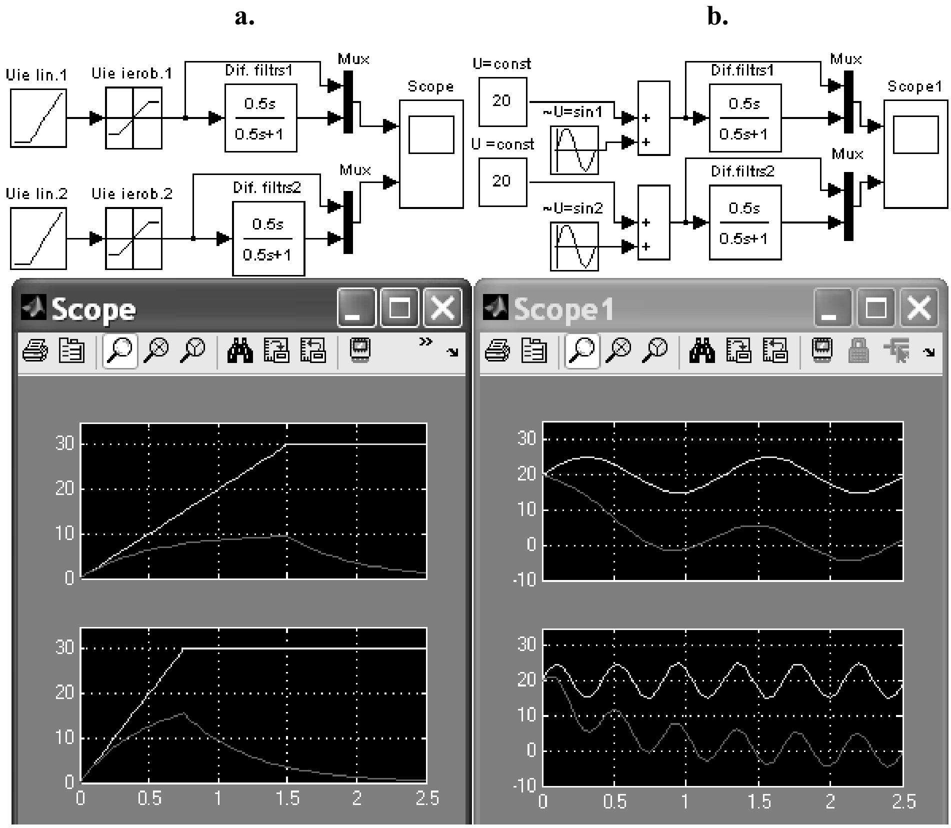
Izvēlamies elektrolītisko kondensatoru ar kapacitāti **C = 50 μF** un potenciometru ar nominālo pretestību **R = 10 kΩ**. Tad diferencēšanas laika konstante **$T_d = C \cdot R = 50 \cdot 10^{-6}\ F \cdot 10 \cdot 10^3\ \Omega = 0.5$ s.**

Noskaidrosim inerciāla diferencējoša posma reakciju uz lineāri augošu, konstantu un svārstīgu ieejas iedarbi. Lai iegūtu diferencējošas CR ķēdes pārejas procesa raksturlīknes, sastādām modelēšanas blokshēmas (3.25. att.).

Lai modelētu diferencējošas CR ķēdes reakciju uz lineāri augošu ieejas spriegumu ar ierobežojumu, no **„Simulink”** bibliotēkas izvēlamies blokus: **„Linear” (U_{ie} lin.1** un **U_{ie} lin.2)** lineāri augoša ieejas sprieguma formēšanai**; „Saturation” (U_{ie} ierob.1** un **U_{ie} ierob.2)** lineāri augošā ieejas sprieguma maksimālā lieluma ierobežošanai**; „Transfer Function” (Dif. filtrs 1** un **Dif. filtrs 2)** diferencējošās ķēdes pārejas procesa modelēšanai; signālu jaucējus **„Mux”** ieejas un izejas raksturlīkņu attēlošanai vienā koordinātu sistēmā un **„Scope”** (3.25.a att.).

Lai modelētu diferencējošas CR ķēdes reakciju uz sinusoidāli svārstīgu ieejas spriegumu papildus izvēlamies blokus: **„Constant” (U = const.),** ieejas signāla konstantās komponentes formēšanai**; „Sine Wave” (~U = sin1** un ~**U = sin2),** ieejas signāla sinusoidālās komponentes formēšanai, un summatoru **„Sum”** (3.25.b att.).

Konfigurējam blokus. Iestatām ieejas spriegumu augšanas ātrumus $\mathbf{v_{Uielin.1}}$ **= 20 V/s** un $\mathbf{v_{Uielin.2}}$ **= 40 V/s**. Maksimālā sprieguma ierobežojums abos gadījumos vienāds ar $\mathbf{U_{ierob.1}}$ **=** $\mathbf{U_{ierob.2}}$ **= 30 V.** Blokos **„Dif.filtrs 1”** un **„Dif.filtrs 2”** ievadām pārvades funkciju (3.37) un tās skaitliskos lielumus: pārvades koeficientu $\mathbf{K_d}$ **= 1** un diferencēšanas laika konstanti $\mathbf{T_d}$ **= 0.5 s.** Blokos **„U = const.”** ievadām ieejas sprieguma konstanto komponenti **U = 20 V,** bet blokos **„~U = sin1”** un **„~U = sin2”** – sinusoidālās komponentes: $\mathbf{U_{1max}}$ **= 5 V,** $\boldsymbol{\omega_1}$ **= 5 rad/s** un $\mathbf{U_{2max}}$ **= 5 V,** $\boldsymbol{\omega_2}$ **= 15 rad/s.**



3.25. att. **Diferencējoša CR filtra modelēšanas blokshēmas un pārejas procesa raksturlīknes $\mathbf{U_{ie}(t)}$ un $\mathbf{U_{iz}(t)}$: a– lineāri augošam ieejas spriegumam ar ierobežojumu 30 V un diviem augšanas ātrumiem ($\mathbf{v_1}$ = 20 V/s, $\mathbf{v_2}$ = 40 V/s); b - svārstīgam ieejas spriegumam ($\mathbf{U_{ie1}}$ = 20 + 5·sin5t, $\mathbf{U_{ie2}}$ = 20 + 5·sin15t)**

Secinājumi

1. Padodot diferencējoša CR filtra ieejā lineāri augošu spriegumu ar ierobežojumu, izejā iegūst zāģveida impulsu, kura maksimālā amplitūda ir proporcionāla sprieguma augšanas ātrumam. Noņemot ierobežojumu, filtra izejā iestatīsies konstanti spriegumi: $\mathbf{U_{izmax1} = T_d \cdot v_{Uie1} = 0.5 \cdot 20 = 10\ V}$ un $\mathbf{U_{izmax2} = T_d \cdot v_{Uie2} = 0.5 \cdot 40 = 20\ V}$.

2. Padodot ieejā svārstīgu spriegumu, kas sastāv no līdzstrāvas komponentes, kurai uzklāta maiņstrāvas komponente, izejā iegūst svārstīgi rimstošu impulsu, kura līdzstrāvas komponente norimst līdz nullei, bet nerimstošās maiņstrāvas komponentes amplitūda un frekvence ir proporcionāla ieejas sprieguma svārstību amplitūdai un frekvencei.

3.9. Automātiskās vadības sistēmas posmu slēgumu īpašības

AVS algoritmisko blokshēmu var sastādīt no raksturīgo posmu dažādiem slēgumiem. Izplatītākie ir virknes, paralēlais un atgriezeniskās saites slēgumi. Posmu ieejas un izejas lielumi ir reālo mainīgo lielumu attēli, kas izteikti kā funkcijas no Laplasa argumenta s. Posma dinamiskās īpašības apraksta pārvades funkcija, kas ir vienāda ar posma izejas un ieejas attēlu attiecību. Lai izpētītu dažādu AVS komponentu slēgumu īpašības, sastādīsim to modelēšanas blokshēmas.

3.9. 1. Virknes slēguma modelēšanas blokshēmas un raksturlīknes

Automātikas pamatu kursā noskaidrojām, ka virknē slēgtu posmu pārvades funkcija ir vienāda ar atsevišķo posmu pārvades funkciju reizinājumu. Pieņemsim, ka vaļēja AVS sastāv no viena bezinerces proporcionālā posma un diviem pirmās kārtas inerciāliem posmiem, kurus vispārīgi apraksta sekojošas pārvades funkcijas:

$$W_1 = K_1; \quad W_2 = \frac{K_2}{T_1 \cdot s + 1}; \quad W_3 = \frac{K_3}{T_2 \cdot s + 1}, \qquad (3.38)$$

kur $\mathbf{K_1, K_2, K_3}$ – posmu pārvades koeficienti;

$\mathbf{T_1, T_2}$ - inerciālo posmu laika konstantes, **s**;

s – Laplasa arguments, $\mathbf{s^{-1}}$.

Posmu virknes slēguma pārvades funkcija:

$$W_v = W_1 \cdot W_2 \cdot W_3 = K_1 \cdot \frac{K_2}{T_1 \cdot s + 1} \cdot \frac{K_3}{T_2 \cdot s + 1} = \frac{K_v}{(T_1 \cdot s + 1)(T_2 \cdot s + 1)}, \quad (3.39)$$

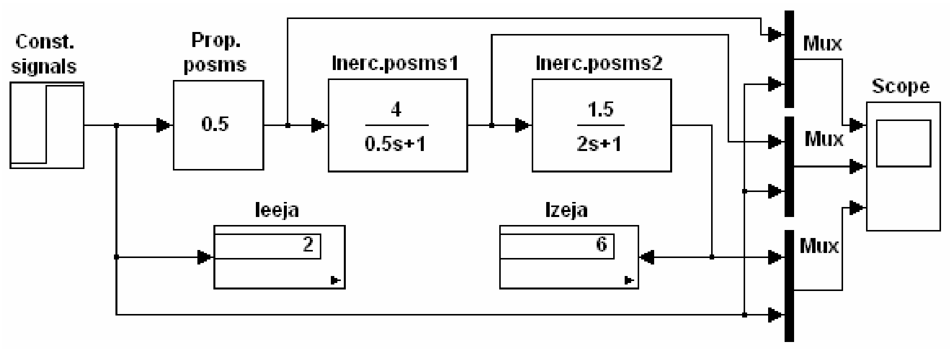
kur $\mathbf{K_v = K_1 \cdot K_2 \cdot K_3}$ – vaļējas AVS pārvades koeficients.

Modelēšanai izvēlamies posmu parametrus, piemēram, $\mathbf{K_1 = 0.5}$; $\mathbf{K_2 = 4}$; $\mathbf{K_3 = 1.5}$ un $\mathbf{T_1 = 0.5}$ **s**; $\mathbf{T_2 = 2}$ **s,** un sastādām modelēšanas blokshēmu virknes slēgumam.

Modelēšanas blokshēmas un raksturlīknes. Noskaidrosim vaļējas automātiskās vadības sistēmas (AVS) ar trim virknē slēgtiem posmiem reakciju uz lēcienveida ieejas signālu $\mathbf{X_{ie} = 2}$ un katra posma iespaidu uz pārejas procesu.

Izvēlamies blokus no **„Simulink”** bibliotēkas: **„Step” (Const. signāls) -** konstanta lēciena ieejas iedarbes formēšanai; **„Slider Gain” (Prop. posms) -** proporcionālā posma modelēšanai; **„Transfer Function” (Inerc. posms1** un **Inerc. posms2)** – pirmās kārtas inerciālo posmu modelēšanai; **„Display” (Ieeja** un **Izeja)-** ieejas un izejas signālu nostabilizējušos lielumu nolasīšanai; **„Mux”** – signālu jaucējus AVS ieejas signāla un katra posma izejas signāla attēla iegūšanai vienā koordinātu sistēmā; **„Scope”** – osciloskopu pārejas procesa raksturlīkņu vizualizācijai.

Konfigurējam AVS posmu virknes slēguma modeļa algoritmiskos blokus. Blokā „**Const. signāls”** ievada $\mathbf{t \geq 0}$, $\mathbf{X_{ie} = 2}$. Aktivizē bloku „**Prop. posms”** un ar slīdni iestata proporcionālā posma pārvades koeficientu $\mathbf{K_1 = 0.5}$. Atver pēc kārtas blokus **„Inerc. posms1”** un „**Inerc. posms2”** un ievada inerciālo posmu pārvades funkciju parametrus: $\mathbf{K_2 = 4}$, $\mathbf{T_1 = 0.5}$ **s** un $\mathbf{K_3 = 1.5}$, $\mathbf{T_2 = 2}$ **s.**

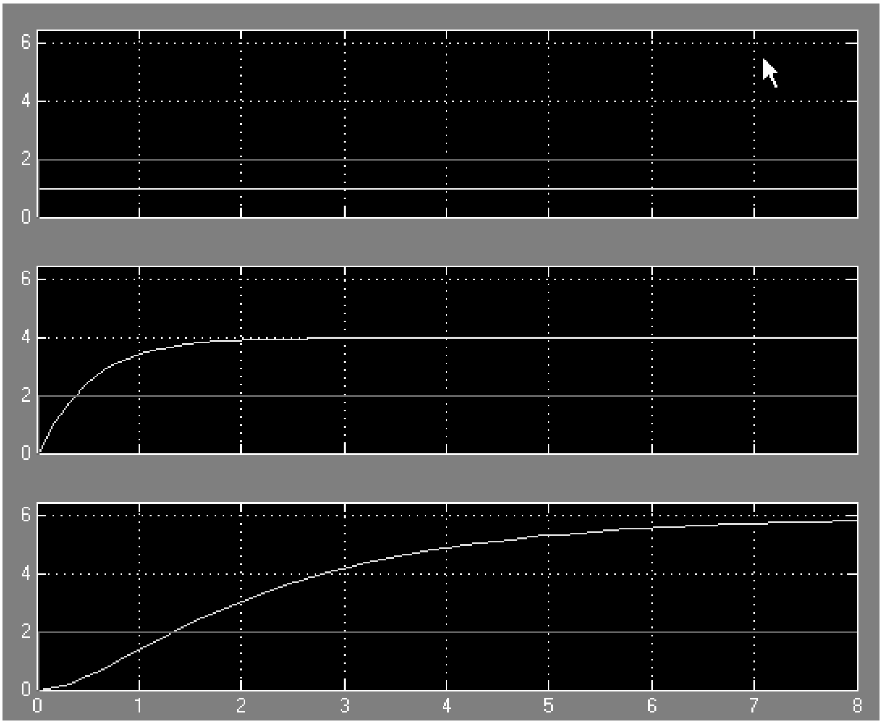


3.26. att. **Vaļējas AVS ar trim virknē slēgtiem posmiem modelēšanas blokshēma, ja ieejas iedarbe mainās lēcienveidā X_{ie} = 2**

Veicot simulāciju, iegūstam katra posma un to virknes slēguma pārejas procesa raksturlīknes (3.27. att.). Ciparu displeji parāda ieejas un izejas signālu nostabilizējušos lielumus. Varam pārbaudīt automātikas pamatos definēto sakarību, ka virknes slēguma pārvades koeficients ir vienāds ar atsevišķo posmu pārvades koeficientu reizinājumu: $\mathbf{X_{izs} = K_1 \cdot K_2 \cdot K_3, \cdot X_{ie} = K_v \cdot X_{ie} = 0.5 \cdot 4 \cdot 1.5 \cdot X_{ie} = 3 \cdot X_{ie} = 3 \cdot 2 = 6}$.

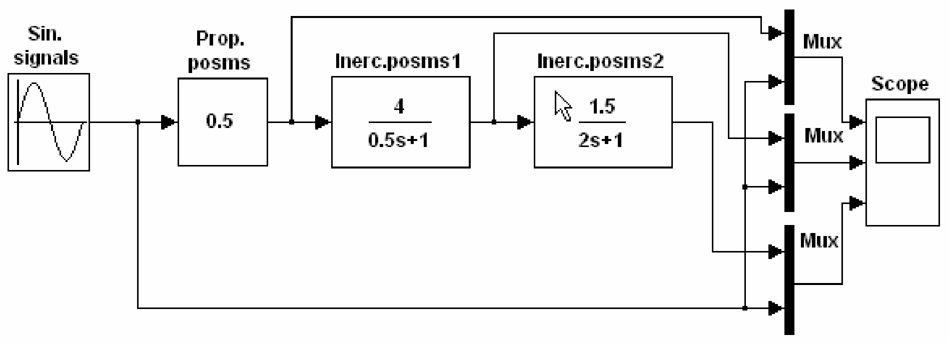
Padodot virknes slēguma ieejā lēcienveida signālu, arī proporcionālā posma izejā signāls izmainās lēcienveida - $\mathbf{X_p = K_1 \cdot X_{ie} = 0.5 \cdot 2 = 1}$. Pirmā inerciālā posma izejas signāls pieaug ar aizkavēšanos, kuru nosaka inerces rādītājs – laika konstante $\mathbf{T_1 = 0.5}$ **s**. Pārejas procesa beigās **1. inerciālā** posma izejas signāls sasniedz maksimālo nostabilizējušos vērtību: $\mathbf{X_{1s} = K_1 \cdot K_2 \cdot X_{ie} = 0.5 \cdot 4 \cdot 2 = 4}$.

2. inerciālais posms vēl vairāk aizkavē ieejas signāla pārvadi, jo tā laika konstante $\mathbf{T_2 = 2}$ **s** ir 4 reizes lielāka par **1. inerciālā posma** laika konstanti $\mathbf{T_1 = 0.5}$ **s.** Pārejas procesa beigās **2. inerciālā posma** izejas signāls, kas vienlaicīgi ir virknes slēguma izejas signāls, sasniedz maksimālo nostabilizējušos vērtību: $\mathbf{X_{2s} = X_{izs} = K_1 \cdot K_2 \cdot K_3 \cdot X_{ie} = 0.5 \cdot 4 \cdot 1.5 \cdot 2 = 6}$. Izejas lieluma pārejas procesa laiku nosaka abu inerciālo posmu laika konstanšu summa. Var pieņemt, ka izejas lieluma pārejas procesa (nostabilizēšanās) laiks: $\mathbf{tp \geq 3 \cdot (T_1 + T_2) = 3 \cdot (0.5 + 2) = 7.5}$ **s.**



3.27. att. **Vaļējas AVS ar virknē slēgtiem posmiem pārejas procesa raksturlīknes $X_{ie}(t)$ un $X_{iz1}(t)$, $X_{iz2}(t)$, $X_{iz3}(t)$ lēcienveida ieejas iedarbei X_{ie} = 2**

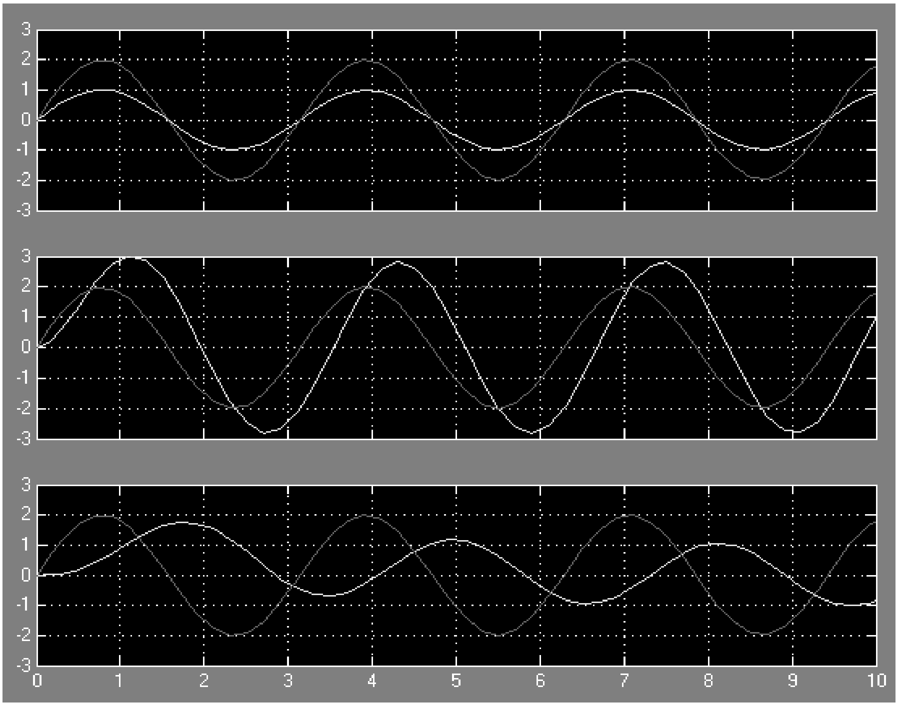
Noskaidrosim posmu virknes slēguma reakciju uz harmonisku (sinusoidālu) ieejas signālu: $\mathbf{X_{ie} = X_{iemax} \cdot \sin \omega t}$. Ieejā uzstādām signālu ģeneratoru **„Sine Wave"** (**Sin. signāls**) (3.28. att.). Bloka parametru logā ievadām: $\mathbf{X_{iemax} = 20}$ un $\mathbf{\omega = 2}$ **rad/s.**



3.28. att. **Vaļējas AVS ar trim virknē slēgtiem posmiem modelēšanas blokshēma sinusoidālai ieejas iedarbei X_{ie} = 20 sin 2t**

Veicot simulāciju, iegūstam katra posma un posmu virknes slēguma pārejas procesa raksturlīknes sinusoidālam ieejas signālam (3.29. att.).

Padodot virknes slēguma ieejā sinusoidālu signālu, proporcionālā posma izejā arī rodas sinusoidāls signāls ar tādu pašu frekvenci un sākuma fāzi. Izejas signāla amplitūdu nosaka tikai ieejas signāla amplitūda un proporcionālā posma pārvades koeficients: $\mathbf{X_{pmax} = K_1 \cdot X_{iemax} = 0.5 \cdot 2 = 1}$. Savādāk uz harmonisku ieejas signālu reaģē inerciāls posms. **1. inerciālā** posma izejas signāls aizkavējas (nobīdās) laikā attiecībā pret ieejas signālu par lielumu $\mathbf{t_{nob1} \approx 0.3\ s}$, kuru nosaka laika konstante $\mathbf{T_1}$. Vienlaicīgi samazinās posma dinamiskais pārvades koeficients $\mathbf{K_{2din}}$ salīdzinājumā ar statisko pārvades koeficientu $\mathbf{K_2 = 4}$. Izejas signāla amplitūda $\mathbf{X_{1max} = 2.7}$ (sk. 2. logu 3.29. att.). Noteiksim **1. inerciālā posma** dinamisko pārvades koeficientu: $\mathbf{K_{2din}} =$ $= \mathbf{X_{1max}/(K_1 \cdot X_{iemax}) = 2.7/(0.5 \cdot 2) = 2.7}$. Tas samazinājies **4/2.7 = 1.48 reizes.**



3.29. att. **Vaļējas AVS ar virknē slēgtiem posmiem pārejas procesa raksturlīknes $X_{ie}(t)$ un $X_{iz1}(t)$, $X_{iz2}(t)$, $X_{iz3}(t)$ sinusoidālai ieejas iedarbei X_{ie} = 20 sin 2t**

2. inerciālais posms vēl vairāk aizkavē ieejas signāla pārvadi, jo tā laika konstante ir **4** reizes lielāka par **1. inerciālā posma** laika konstanti**.** To uzskatāmi parāda virknes slēguma ieejas un izejas signālu oscilogrammu salīdzinājums (sk. 3. logu 3.29. att.). Izejas signāla nobīdes laiks $\mathbf{t_{nobiz} \approx 1\ s}$ un izejas signāla amplitūda – $\mathbf{X_{izmax} = 1}$ ir **6 reizes** mazāka salīdzinājumā ar konstantu signālu.

Secinājums. Izejas signāla amplitūda samazinās, bet nobīdes laiks palielinās, palielinoties ieejas signāla frekvencei un AVS posmu inercei.

3.9.2. Paralēlā slēguma modelēšanas blokshēmas un raksturlīknes

Paralēlajā slēgumā visu posmu ieejās tiek padots viens un tas pats signāls. Tā kā katrs posms uz to reaģē dažādi, tad posmu izejās iegūst atšķirīgus signālus, kurus saskaita ar summatoru. Paralēlā slēguma pārvades funkcija ir vienāda ar atsevišķo paralēli savienoto posmu pārvades funkciju summu. Pieņemsim, ka AVS vadības iekārta sastāv no trim paralēli slēgtiem posmiem – proporcionālā, integrējošā un diferencējošā posma, kurus apraksta sekojošas pārvades funkcijas:

$$W_1 = K_p; \quad W_2 = \frac{1}{T_i \cdot s}; \quad W_3 = \frac{T_d \cdot s}{T_d \cdot s + 1}, \tag{3.40}$$

kur $\mathbf{K_p}$– proporcionālā posma pārvades koeficients;

$\mathbf{T_i}$ – integrēšanas laika konstante, **s;**

$\mathbf{T_d}$ - diferencēšanas laika konstante, **s**.

Posmu paralēlā slēguma pārvades funkcija:

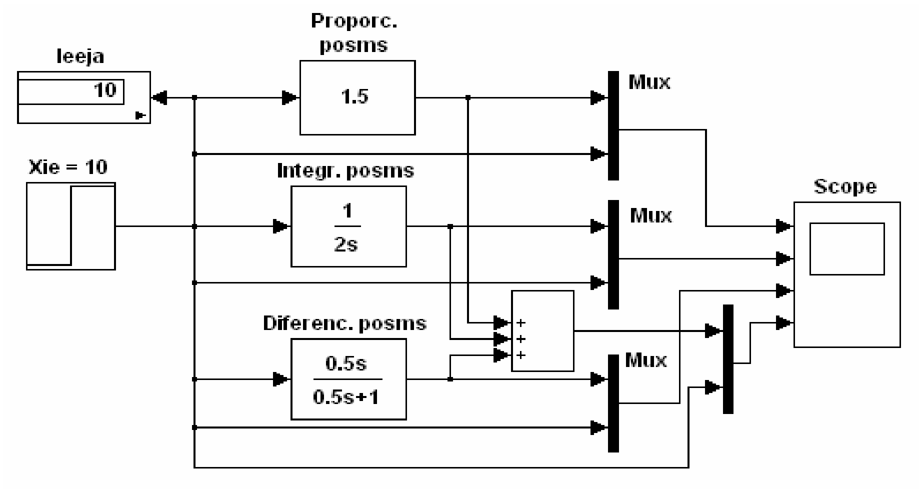
$$W_p = W_1 + W_2 + W_3 = K_p + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{T_d \cdot s + 1}. \tag{3.41}$$

Pārvades funkcija (3.41) apraksta vienkāršotu idealizētu proporcionāli integrālo-diferenciālo **(PID)** vadības iekārtu, kas formē vadības signālu proporcionāli ieejas signāla amplitūdai, integrālim no tā un izmaiņas ātrumam.

<u>Modelēšanas blokshēmas un raksturlīknes</u>. Modelēšanai izvēlamies posmu parametrus: $\mathbf{K_p = 1.5; T_i = 2\ s; T_d = 0.5\ s}$ un sastādām modelēšanas blokshēmu posmu paralēlajam slēgumam (3.30. att.). Proporcionālā posma modelēšanai izvēlamies bloku **„Slider Gain"**, kurā ievadām pārvades koeficientu **0.5.** Integrējošā un diferencējošā posma modelēšanai izvēlamies blokus **„Transfer Function"**, kurus konfigurējam atbilstoši attiecīgo posmu algoritmiem. Ieejā uzstāda bloku **„Step"** un **„Display",** lai formētu lēcienveida signālu $\mathbf{X_{ie} = 10}$ un nolasītu tā skaitlisko vērtību.

Paralēlā slēguma izejas signālu iegūst ar summatoru, bet lai iegūtu ieejas un izejas raksturlīkņu attēlus vienā koordinātu sistēmā, izmanto multipleksorus **„Mux".**

Simulācijas rezultāti parādīti 3.31. attēlā. Uz lēcienveida ieejas iedarbi vispirms reaģē proporcionālais un diferencējošais posms (**a, c** un **d** logi). Tad sāk darboties integrējošā ķēde (**b** un **d** logi), kas likvidē regulēšanas statisko kļūdu.

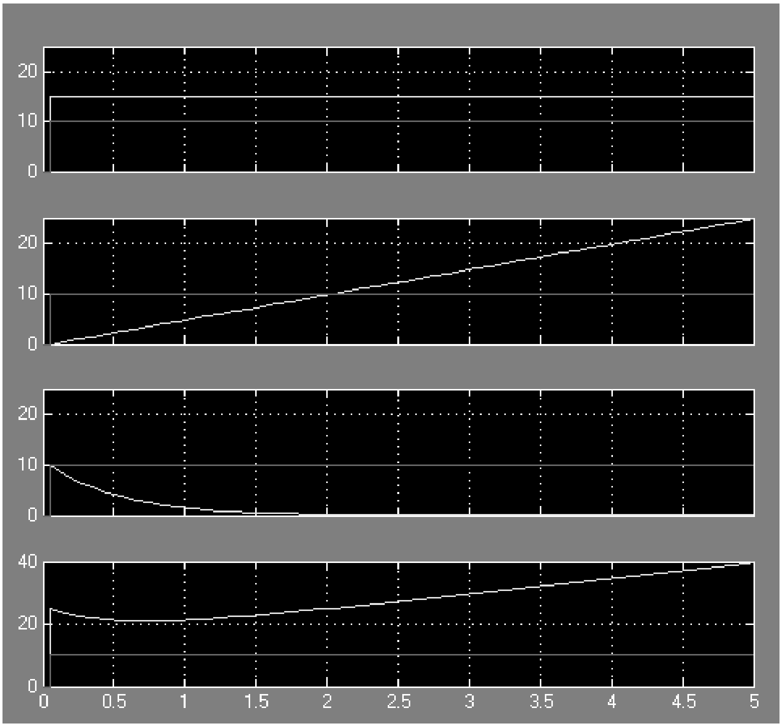


3.30. att. **AVS vadības iekārtas ar trim paralēli slēgtiem posmiem algoritmiskā blokshēma lēcienveida ieejas iedarbei X_{ie} = 10** a.

b.

c.

d.



3.31. att. **AVS vadības iekārtas ar paralēli slēgtiem posmiem pārejas procesa raksturlīknes: a – proporcionālajam posmam; b – integrējošam posmam; c – diferencējošam posmam; d – posmu paralēlajam slēgumam**

3.9.3. Slēgtas AVS ar atgriezenisko saiti pārejas procesa raksturlīknes

Dažādu tehnoloģisko parametru (temperatūras, līmeņa, spiediena, ātruma u.c.) automātiskai stabilizācijai izmanto slēgtas AVS ar negatīvu atgriezenisko saiti.

Atgriezeniskā saite ir negatīva, ja novirzes jeb kļūdas signāls formējas kā ieejas signāla un atgriezeniskās saites signāla starpība. Negatīva atgriezeniskā saite paaugstina sistēmas stabilitāti un precizitāti, tādēļ to izmanto visās slēgtajās tehnoloģisko iekārtu automātiskās vadības sistēmās.

Apskatīsim slēgtu AVS, kuras ieejā tiek padota **lēcienveida iedarbe** $\mathbf{X_{ie}}$. Tā sastāv no **vadības iekārtas** ar pārvades funkciju $\mathbf{W_v}$, **izpildiekārtas** ar pārvades funkciju $\mathbf{W_{ik}}$, **vadības objekta** ar pārvades funkciju $\mathbf{W_{obj}}$ un atgriezeniskās saites ar pārvades funkciju $\mathbf{W_{as}}$.

Uz vadības objektu darbojas **perturbācija X_p**, kas tieši iespaido izpildiekārtas regulējošo iedarbi $\mathbf{X_r}$ un izmaina vadības objekta izejas lielumu $\mathbf{X_{iz}}$.

Slēgtas AVS ar negatīvu atgriezenisko saiti pārejas procesu, ņemot vērā perturbāciju iespaidu, apraksta pārvades funkcija:

$$\Phi = \frac{X_{iz}(s)}{X_{ie}(s)} = \frac{W - \dfrac{X_p(s)}{X_{ie}(s)} \cdot W_{obj}}{1 + W_{as} \cdot W}, \qquad (3.42)$$

kur $\mathbf{X_p(s)}$ – perturbējošās iedarbes uz vadības objektu attēls;

$\mathbf{X_{ie}(s)}$ un $\mathbf{X_{iz}(s)}$ – AVS ieejas iedarbes un izejas lieluma attēls;

$\mathbf{W = W_v \cdot W_{ik} \cdot W_{obj}}$ – vaļējas AVS pārvades funkcija;

$\mathbf{W_{as}}$ – atgriezeniskās saites pārvades funkcija.

Pārvades funkcija (3.42) apraksta slēgtas AVS uzspiesto kustību, kam par iemeslu ir perturbācija X_p, kura izraisa vadības objekta izejas lieluma $\mathbf{X_{iz}}$ novirzi no uzdotā lieluma $\mathbf{X_{iz0}}$. Perturbācija samazina slēgtas AVS pārvades funkciju $\mathbf{\Phi}$ un līdz ar to pasliktina tās darbības kvalitāti.

Slēgtas AVS modelēšanai izvēlamies PID vadības iekārtu ar pārvades funkciju:

$$W_v = K_p + \frac{1}{T_i \cdot s} + \frac{T_d \cdot s}{T_d \cdot s + 1} = 6 + \frac{1}{7 \cdot s} + \frac{2 \cdot s}{2 \cdot s + 1}, \qquad (3.43)$$

kur $\mathbf{K_p = 6}$ – proporcionālās ķēdes pārvades koeficients;

$\mathbf{T_i = 7}$ – integrējošās ķēdes laika konstante, **s**;

$\mathbf{T_d = 2}$ – diferencējošās ķēdes laika konstantes, **s**.

Izvēlamies bezinerces atgriezenisko saiti ar pārvades funkciju: $\mathbf{W_{as} = 0.35}$.

Vadības objekts ir pirmās kārtas inerciāls posms ar pārvades funkciju:

$$W_{obj} = \frac{K_{obj}}{T_{obj} \cdot s + 1} = \frac{5}{1.2 \cdot s + 1}, \qquad (3.44)$$

kur $\mathbf{K_{obj} = 5}$ – vadības objekta pārvades koeficients;

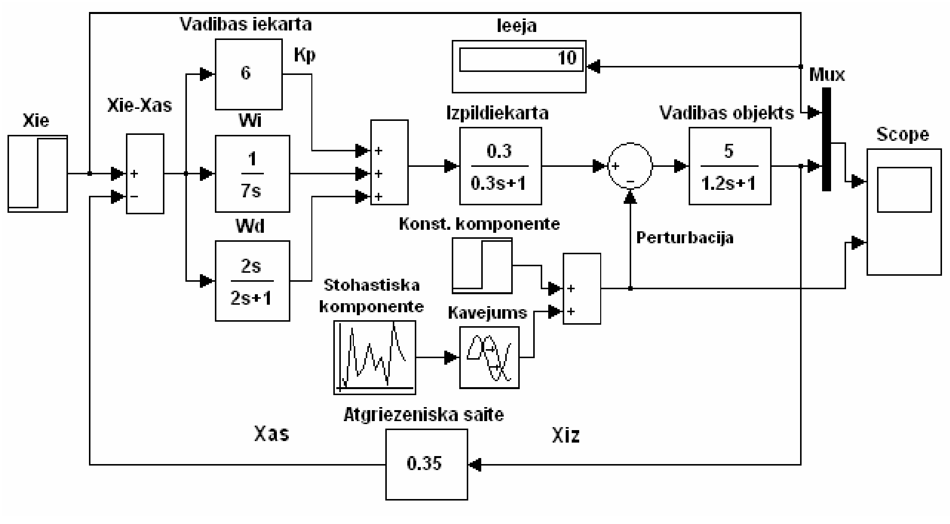
T_{obj}**=1.2 –** vadības objekta laika konstante, **s;**

s – Laplasa arguments, $\mathbf{s^{-1}}$.

<u>Modelēšanas blokshēmas un raksturlīknes</u>. Sastādām slēgtas AVS ar negatīvu atgriezenisko saiti modelēšanas blokshēmu no „Simulink” blokiem (3.32. att.).

Vadības iekārtai izmantojam iepriekš sastādīto modeli (3.30. att.). Izpildiekārtas un vadības objekta modelēšanai izvēlamies blokus **„Transfer Function”**, kurus konfigurējam atbilstoši attiecīgo posmu algoritmiem. Ieejā uzstādām bloku **„Step”** un **„Display”** lēcienveida signāla $\mathbf{X_{ie}}$ **= 10** formēšanai un tā vērtības nolasīšanai.

Atgriezenisko saiti imitējam ar bloku **„Slider Gain”**, ievadot tajā koeficientu **0.35**.



3.32. att. **Slēgtas AVS ar negatīvu atgriezenisko saiti algoritmiskā blokshēma**

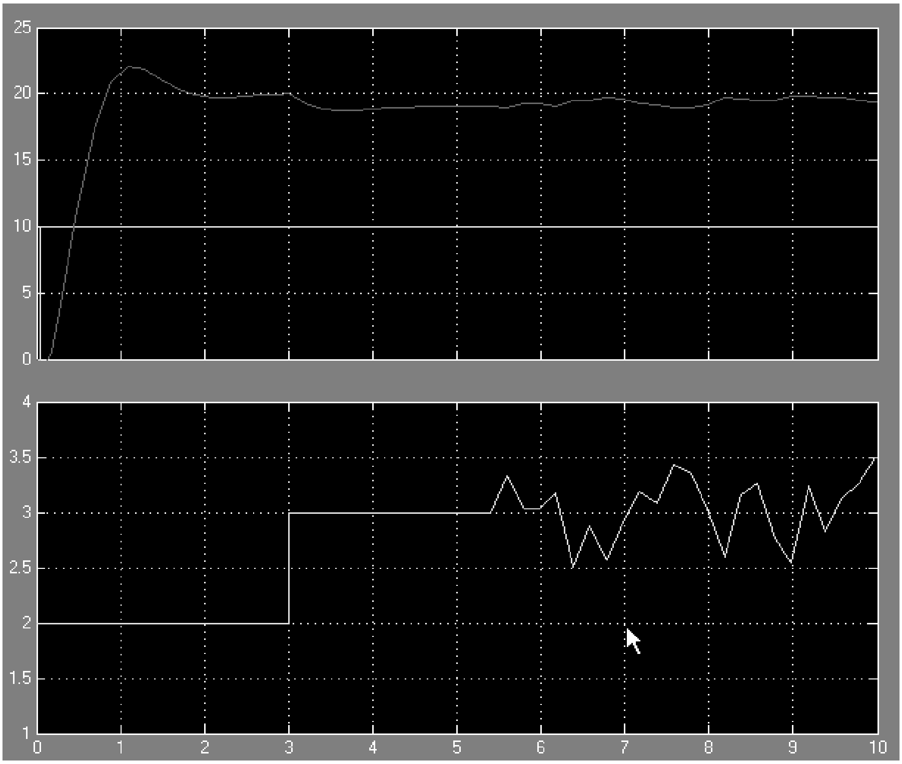
Lai modelētu perturbāciju, kas darbojas vadības objekta ieejā, izvēlamies 4 blokus: **„Step”** – divu līmeņu konstanta lēciena perturbācijas (**Konst. komponente**) formēšanai; **„Random Number”**- gadījuma signāla ar normālo (Gausa) sadalījumu

(**Stohastiska komponente**) formēšanai; **„Transport Delay"** (**Kavējums**) – gadījuma signāla aizkavēšanai; **„Sum"** – konstantās un gadījuma komponentes summēšanai.

Lai iegūtu slēgtas AVS pārejas procesa raksturlīkni $\mathbf{X_{iz}(t)}$ pie lēcienveida ieejas iedarbes $\mathbf{(t > 0, X_{ie} = 10)}$ (3.33.a att.) atkarībā no mainīgas perturbācijas $\mathbf{(0 < t \le 3\ s}$, $\mathbf{X_p = 2;\ 3 < t \le 5.5\ s,\ X_p = 3;\ t > 5.5\ s,\ X_p = 3 + Fn}$ **(±0.5**) (3.33.b att.) attiecīgi konfigurējam perturbācijas formēšanas blokus**. Fn (±0.5)** ir normālā sadalījuma funkcija ar signāla izkliedes amplitūdu **±0.5**. Raksturlīkne $\mathbf{X_{iz}(t)}$ parāda, ka palielinoties perturbācijai, regulējamais parametrs novirzās no uzdotās vērtības $\mathbf{X_{iz0} = 20.}$ Šo novirzi pakāpeniski samazina integrējošā ķēde. Redzams, ka vadības sistēma labi kompensē stohastiskās komponentes iespaidu, jo izejas parametra svārstības ir mazākas par 5%.

a.

b.



3.33. att. **Slēgtas AVS ar PID vadības iekārtu pārejas procesa raksturlīknes: a– ieejas un izejas signāliem** X_{ie}**=f(t),** X_{iz}**=f(t); b – perturbācijai** X_p**=f(t)**

4. AVS VADĪBAS IEKĀRTU IZVĒLE UN PĀREJAS PROCESU MODELĒŠANA

Tehnoloģisko iekārtu vadību mūsdienās realizē izmantojot programmējamos kontrollerus, kuru uzdevums ir kontrolēt un automātiski regulēt tehnoloģiskajās iekārtās notiekošos procesus.

Tehnoloģisko iekārtu vadībai un tajās notiekošo procesu regulēšanai mūsdienās izmanto:

- loģiskos kontrollerus (regulējošā iedarbe uz objektu tiek padota periodiski);
- regulējošos kontrollerus (iedarbe uz objektu tiek regulēta nepārtraukti);
- speciālos impulsregulēšanas kontrollerus (iedarbe uz objektu tiek regulēta impulsveidīgi).

Tehnoloģisko iekārtu nepārtrauktai vadībai visplašāk tiek izmantoti regulējošie kontrolleri. Apskatīsim to darbības likumus, izvēles nosacījumus un pārejas procesa raksturlīkņu modelēšanu Windows vidē.

4.1. Proporcionālā regulatora darbības algoritms un īpašības

Proporcionālais regulēšanas likums ir visvienkāršākais. **P – regulatora** izejas signāls mainās tieši proporcionāli ieejas signālam. Tā dinamiskās īpašības novērtē pēc reakcijas uz lēcienveida ieejas iedarbi. Padodot lēcienveida signālu ideāla **P – regulatora** ieejā, arī izejas signāls mainās lēcienveidīgi bez aizkavēšanas.

P – regulatora galvenais trūkums ir statiskā kļūda, kuru rada ciešā saite starp ieejas un izejas lielumiem. Statisko kļūdu var samazināt, palielinot **P – regulatora** statisko pārvades koeficientu (jutību), bet to nav iespējams pilnībā likvidēt.

P–regulatora darbību kā statiskā, tā dinamiskā režīmā apraksta proporcionālā posma algoritmi:

- oriģinālvienādojums – $X_{iz} = K_p \cdot X_{ie}$;
- operatorvienādojums – $X_{iz}(s) = K_p \cdot X_{ie}(s)$; (4.1)
- pārvades funkcija – $W_p = X_{iz}(s) / X_{ie}(s) = K_p$,

kur $\mathbf{K_p = X_{iz}/X_{ie}}$ – **P – regulatora** statiskais pārvades (pastiprinājuma) koeficients.

Jo lielāks koeficients $\mathbf{K_p}$, jo mazāka P – regulatora radītā statiskā kļūda.

Automātikas pamatu kursā noskaidrojām, ka AVS ar **P – regulatoru** relatīvo statisko kļūdu pēc ieejas iedarbes nosaka sistēmas posmu jutība:

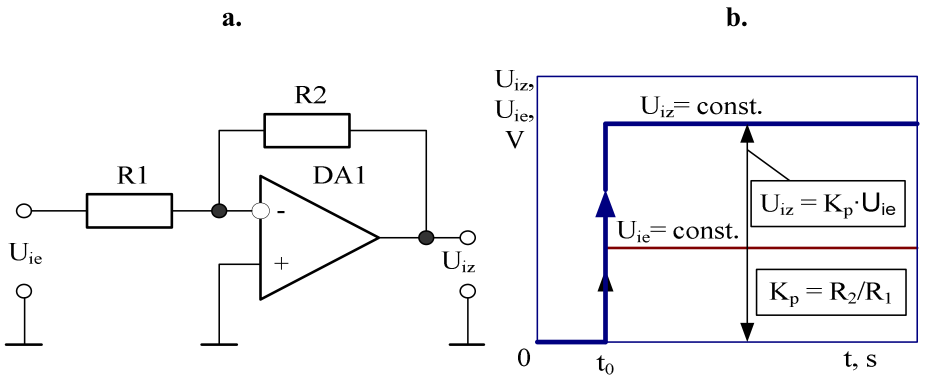
$$\boldsymbol{\varepsilon = 1 / (1 + K_p \cdot K_s) \cdot 100\%,} \qquad (4.2)$$

kur $\mathbf{K_s}$ –tehnoloģiskās sistēmas pārvades koeficients.

Ja $\mathbf{K_s}$ uzdots kā konstants lielums, tad AVS ar **P – regulatoru** statisko kļūdu var samazināt, palielinot $\mathbf{K_p}$. Taču palielinot $\mathbf{K_p}$, AVS kļūst svārstīgāka un līdz ar to nestabilāka. Projektējot AVS ar P – regulatoru, koeficienta $\mathbf{K_p}$ izvēles kritērijs parasti ir pieļaujamā statiskā kļūda (4.2). Pēc tam pārbauda AVS stabilitāti un darbības kvalitāte. Ja šie rādītāji neapmierina prasības, jāveic papildus pasākumi AVS darbības stabilitātes uzlabošanai, piemēram, sistēmu papildina ar virknes vai paralēlās korekcijas ķēdēm.

Apskatīsim elektronisku **P**–regulatoru, kas sastāv no operacionālā pastiprinātāja **DA1**, ieejas rezistora **R1** un atgriezeniskās saites rezistora **R2** (4.1.a att). Tā pārvades koeficientu var izteikt ar operacionālā pastiprinātāja ieejas ķēdes pretestību $\mathbf{R_1}$ un atgriezeniskās saites pretestību $\mathbf{R_2}$:

$$\mathbf{K_p = U_{iz} / U_{ie} = R_2/R_1.} \qquad (4.3)$$



4.1. att. **Elektroniskais P – regulators: a – elektriskā principshēma; b – ieejas un izejas spriegumu pārejas procesa raksturlīknes: U_{ie} = f(t), U_{iz} = f(t)**

Padodot elektroniskā P – regulatora ieejā lēcienveida spriegumu $\mathbf{U_{ie}}$ **= const**, arī izejas spriegums $\mathbf{U_{iz}}$ pieaug lēcienveidīgi bez aizkavēšanās, jo operacionālā pastiprinātāja ārējā ķēdē ir tikai aktīvie elementi.

4.2. Integrālā regulatora darbības algoritms un īpašības

I – regulatora izejas signāls mainās tieši proporcionāli integrālim no ieejas signāla. Tā dinamiskās īpašības novērtē pēc reakcijas uz lēcienveida ieejas iedarbi. Padodot lēcienveida signālu ideāla **I – regulatora** ieejā, tā izejas signāls pieaug vienmērīgi ar konstantu ātrumu. Tātad **I – regulatoram** piemīt visas integrējoša posma īpašības un starp tā ieejas un izejas signāliem nav statiskas sakarības. Tādēļ **I – regulatoru** sauc arī par **astatisku regulatoru**.

I – regulators ir ievērojami lēndarbīgāks par **P – regulatoru**, taču tam piemīt būtiski svarīga pozitīva īpašība – tas **likvidē statisko kļūdu** un līdz ar to **nodrošina augstu regulēšanas precizitāti.**

I – regulatora kā integrējoša posma **oriģinālvienādojumu, operatorvienādojumu** un **pārvades funkciju** var uzrakstīt divējādi:

$$X_{iz}(t) = \frac{1}{T_i} \cdot \int_0^t X_{ie}(t)dt \quad \text{vai} \quad X_{iz}(t) = k_i \cdot \int_0^t X_{ie}(t)dt\,; \tag{4.4}$$

$$X_{iz}(s) = \frac{X_{ie}(s)}{T_i s} \quad \text{vai} \quad X_{iz}(s) = k_i \cdot \frac{X_{ie}(s)}{s}\,; \tag{4.5}$$

$$W_I(s) = \frac{X_{iz}(s)}{X_{ie}(s)} = \frac{1}{T_i \cdot s} \quad \text{vai} \quad W_I(s) = \frac{X_{iz}(s)}{X_{ie}(s)} = \frac{k_i}{s}\,, \tag{4.6}$$

kur $\mathbf{T_i}$ – integrācijas laika kostante, kas raksturo izejas signāla augšanas inerci, **s**;

$\mathbf{k_i = (dX_{iz}/dt) / X_{ie}}$ – ātruma koeficients, kas izsaka izejas signāla izmaiņas ātruma atkarību no ieejas signāla lieluma , $\mathbf{s^{-1}}$.

Lēcienveida ieejas iedarbes gadījumā **($t < 0$, $X_{ie} = 0$; $t \geq 0$, $X_{ie} = const$)**, izejas signāls X_{iz} pieaug proporcionāli laikam **t**, tad $k_i = (X_{iz} / t) / X_{ie}$, no kurienes iegūstam ekvivalento pārvades koeficientu: $k_e = X_{iz} / X_{ie} = k_i \cdot t$.

I – regulatora relatīvā statiskā kļūda: $\varepsilon = 1/ (1 + k_i \cdot t \cdot K_s) \cdot 100\%$, kur K_s – tehnoloģiskās sistēmas pārvades koeficients. Tātad augot laikam **t**, **I – regulators** pakāpeniski likvidē regulēšanas statisko kļūdu.

Elektroniskais **I – regulators** sastāv no operacionālā pastiprinātāja, kura ieejas ķēdē slēgts rezistors **R1**, bet atgriezeniskajā saitē – kondensators **C1** (4.2.a att.).

Elektroniskajā **I – regulatorā** notiekošos pārejas procesus apraksta sekojošs **oriģinālvienādojums, operatorvienādojums** un **pārvades funkcija**:

$$U_{iz}(t) = \frac{1}{T_i} \cdot \int_0^t U_{ie}(t)dt , \tag{4.7}$$

$$U_{iz}(s) = \frac{U_{ie}(s)}{T_i s} , \tag{4.8}$$

$$W_I = \frac{U_{iz}(s)}{U_{ie}(s)} = \frac{1}{T_i \cdot s} , \tag{4.9}$$

kur $U_{ie}(t)$ un $U_{iz}(t)$ – ieejas un izejas spriegumi kā laika t funkcijas;

$U_{ie}(s)$ un $U_{iz}(s)$ – ieejas un izejas spriegumu attēli s – koordinātu sistēmā;

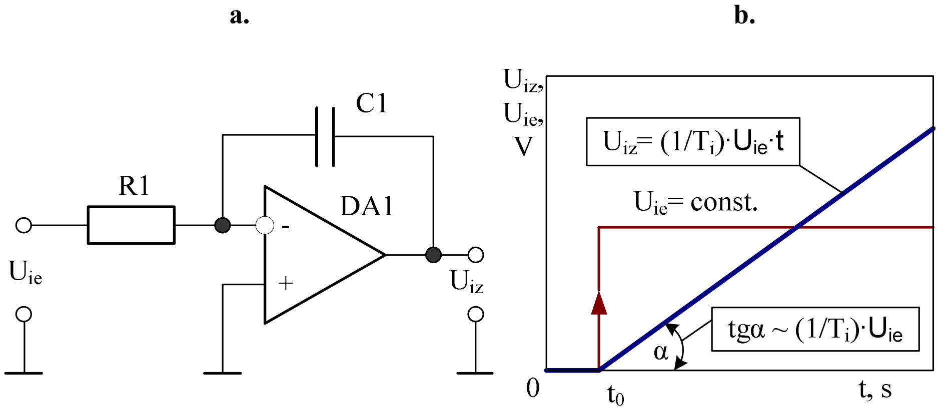
$T_i = C_1 \cdot R_1$ **($F \cdot \Omega = s$) -** integrācijas laika konstante.

Atrisinot vienādojumu **(4.7)** pie nulles sākuma nosacījumiem un lēcienveida ieejas sprieguma izmaiņas, iegūst izejas sprieguma pārejas procesa raksturlīknes (4.2.b att.) analītisko izteiksmi:

$$U_{iz} = \frac{U_{ie} \cdot t}{T_i} \tag{4.10}$$

Padodot elektroniskā **I – regulatora** ieejā lēcienveida spriegumu U_{ie} = **const**, notiek pakāpeniska kondensatora **C1** uzlāde caur rezistoru **R1** ar laika konstanti

$T_i = C_1 \cdot R_1$. Lai **I – regulators** nodrošinātu kvalitatīvu pārejas procesu vadības objektā, integrācijas laika konstantei T_i jābūt atbilstoši saskaņotai ar vadības objekta laika konstanti T_{obj}. Lielākai vadības objekta laika konstantei T_{obj}, izvēlas proporcionāli lielāku integrācijas laika konstanti T_i. Vadības iekārtu parametru izvēles kritēriji tiks apskatīti šīs nodaļas izklāsta turpinājumā.



4.2. att. **Elektroniskais I – regulators: a – elektriskā principshēma; b – pārejas procesa raksturlīkne $U_{iz}=U_{ie} \cdot t/T_i$ lēcienveida ieejas iedarbei U_{ie}=const**

I - regulatoru vienu pašu praksē izmanto reti, jo tas padara sistēmu lēndarbīgu un svārstīgu. I – regulēšanas likuma galvenais uzdevums ir novērst statisko kļūdu, tādēļ to parasti kombinē ar citiem regulēšanas veidiem, piemēram, ar proporcionālo.

4.3. Proporcionāli integrālā regulatora darbība un īpašības

Apvienojot **proporcionālo** un **integrālo** vadības likumu iegūst **PI – regulatoru**. **PI – regulatora** izejas signāls mainās tieši proporcionāli ieejas signālam un integrālim no tā. Tā dinamiskās īpašības novērtē pēc reakcijas uz lēcienveida ieejas iedarbi.

Padodot lēcienveida signālu ideāla **PI – regulatora** ieejā, vispirms sāk darboties proporcionālā ķēde, kas formē izejas signāla lēcienveida komponenti. Pēc tam sāk darboties integrējošā ķēde, kura ir daudz lēndarbīgāka un pakāpeniski likvidē regulēšanas statisko kļūdu.

PI – regulatoram piemīt proporcionālā un integrālā regulatora pozitīvās īpašības – liela ātrdarbība un augsta precizitāte. Tādēļ **PI – regulators** ir augstākas klases regulators salīdzinājumā ar iepriekš apskatītajiem un nodrošina augstāku regulēšanas procesa stabilitāti un kvalitāti.

PI – regulatora struktūru veido proporcionālā un integrējošā posma paralēlais slēgums. Izmantojot posmu paralēlā slēguma īpašības, iegūstam **PI – regulatora oriģinālvienādojumu:**

$$X_{iz}(t) = K_p \cdot X_{ie}(t) + \frac{1}{T_i} \cdot \int_0^t X_{ie}(t)dt, \qquad (4.11)$$

operatorvienādojumu:

$$X_{iz}(s) = K_p \cdot X_{ie}(s) + \frac{X_{ie}(s)}{T_i \cdot s} \qquad (4.12)$$

un **pārvades funkciju:**

$$W_{PI}(s) = \frac{X_{iz}(s)}{X_{ie}(s)} = K_p + \frac{1}{T_i \cdot s}, \qquad (4.13)$$

kur $\mathbf{T_i}$ – integrējošās ķēdes laika kostante, **s**;

$\mathbf{X_{iz}(s) / X_{ie}(s)}$ – izejas un ieejas signālu attēli;

$\mathbf{K_p = X_{iz} / X_{ie}}$ – proporcionālās ķēdes pārvades koeficients.

Lēcieveida ieejas iedarbes gadījumā ($\mathbf{t < 0, X_{ie} = 0; t \geq 0, X_{ie} = const}$), izejas signāls $\mathbf{X_{iz}}$ pieaug lēcienveidīgi līdz lielumam: $\mathbf{X_{iz1} = K_p \cdot X_{ie}}$, pēc tam turpina palielināties proporcionāli laikam **t**:

$$\mathbf{X_{iz} = K_p \cdot X_{ie} + (1/T_i) \cdot X_{ie} \cdot t.} \qquad (4.14)$$

PI – regulatora relatīvā statiskā kļūda:

$$\boldsymbol{\varepsilon = 1/(1+ K_p \cdot 1/ T_i \cdot t \cdot K_s) \cdot 100\%}, \qquad (4.15)$$

kur K_s –tehnoloģiskās sistēmas pārvades koeficients.

Tātad augot laikam **t**, **PI – regulators**, tāpat kā **I – regulators** pakāpeniski likvidē regulēšanas statisko kļūdu.

Elektroniskais **PI – regulators** sastāv no operacionālā pastiprinātāja, ieejas rezistora **R1**, atgriezeniskās saites kondensatora **C1** un rezistora **R2** (4.3.a att.).

Dinamiskos procesus elektroniskajā **PI – regulatorā** apraksta pārvades funkcija:

$$W_{PI}(s) = \frac{U_{iz}(s)}{U_{ie}(s)} = K_p + \frac{1}{T_i \cdot s} = K_p \cdot \frac{T_{iz} \cdot s + 1}{T_{iz} \cdot s}, \qquad (4.16)$$

kur $K_p = R_2 / R_1$ – proporcionālās ķēdes pārvades koeficients;

$T_i = C_1 \cdot R_1$ – integrējošās ķēdes laika konstante, s;

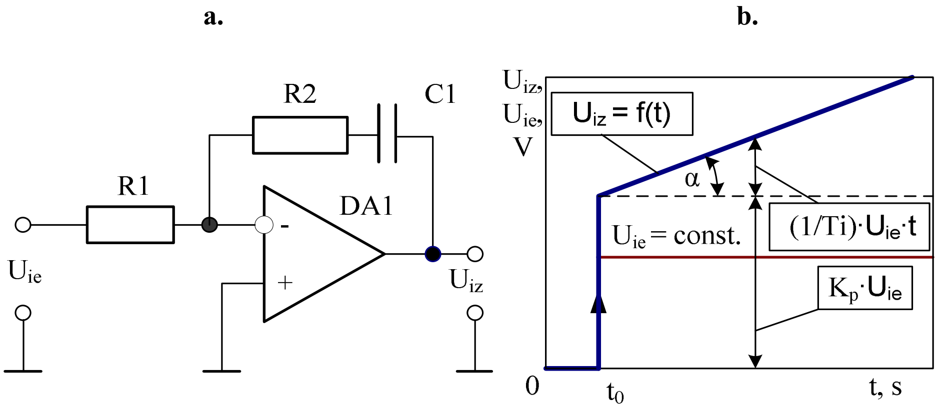
$T_{iz} = T_i \cdot K_p = C_1 \cdot R_1 \cdot R_2/R_1 = C_1 \cdot R_2$ – izodroma laiks, s;

K_p un T_{iz} ir **PI – regulatora** iestatīšanas parametri. Mainot K_p, jāmaina arī T_{iz}.

Padodot elektroniskā **PI – regulatora** ieejā lēcienveida spriegumu **($t < 0$, $U_{ie} = 0$; $t \geq 0$, $U_{ie} = const.$)**, vispirms sāk darboties proporcionālā ķēde, kas formē izejas sprieguma lēcienveida komponenti **($U_{iz1} = K_p \cdot U_{ie}$).** Pēc tam sāk darboties integrējošā ķēde, kura pakāpeniski paaugstina izejas spriegumu proporcionāli laikam t (4.3.b att.):

$$\boldsymbol{U_{iz} = K_p \cdot U_{ie} + (1/T_i) \cdot U_{ie} \cdot t.} \qquad (4.17)$$

Integrējošā ķēde pakāpeniski likvidē regulēšanas statisko kļūdu.



4.3. att. **Elektroniskais PI – regulators: a – elektriskā principshēma; b – pārejas procesa raksturlīknes $U_{ie} = f(t)$, $U_{iz} = f(t)$**

4.4. Proporcionāli diferenciālā regulatora darbība un īpašības

Tradicionāli proporcionāli diferenciālo **PD-regulatoru** realizē kā proporcionālā posma un diferencējošā posma paralēlo slēgumu.

Ideālu diferencējošo posmu nav iespējams realizēt. Tādēļ apskatīsim reālu proporcionāli diferenciālo inerciālo **PD – regulatoru**, kura darbību apraksta sekojošs **oriģinālvienādojums:**

$$T_d \cdot \frac{dX_{iz}(t)}{dt} + X_{iz}(t) = K_p \cdot X_{ie}(t) + \left(K_p + 1\right) \cdot T_d \cdot \frac{dX_{ie}(t)}{dt}; \qquad (4.18)$$

operatorvienādojums:

$$X_{iz}(s) \cdot (T_d \cdot s + 1) = K_p \cdot X_{ie}(s) + (K_p + 1) \cdot T_d \cdot s \cdot X_{ie}(s); \qquad (4.19)$$

un **pārvades funkcija:**

$$W_{PD} = \frac{X_{iz}(s)}{X_{ie}(s)} = K_p + \frac{T_d \cdot s}{T_d \cdot s + 1}, \qquad (4.20)$$

kur $K_p = X_{iz}/ X_{ie}$ – proporcionālā posma pārvades koeficients;

T_d – diferencēšanas laika konstante, **s**.

PD – regulators formē apsteidzošu iedarbi uz vadības objektu. Padodot tā ieejā lēcienveida iedarbi, izejā rodas **impulsveida signāls**, kura amplitūda formējas no **proporcionālās** ķēdes un **diferencējošās** ķēdes signālu summas, jo tās savienotas paralēli.

Ņemot par pamatu izteiksmi (4.20), sastādīsim **elektroniska PD – regulatora** pārvades funkciju:

$$W_{PD} = \frac{U_{iz}(s)}{U_{ie}(s)} = K_p + \frac{T_d \cdot s}{T_d \cdot s + 1} \tag{4.21}$$

Iznesot proporcionālās ķēdes pārvades koeficientu $\mathbf{K_p}$ pirms iekavām un veicot matemātiskus pārveidojumus, pārvades funkciju (4.21) uzrakstām sekojošā formā:

$$W_{PD}(s) = K_p \cdot \frac{T \cdot s + 1}{T_d \cdot s + 1}, \tag{4.22}$$

kur $\mathbf{T = T_d + T_d/K_p = T_d(1+1/K_p)}$ – koriģētā diferencēšanas laika konstante*,* **s**;

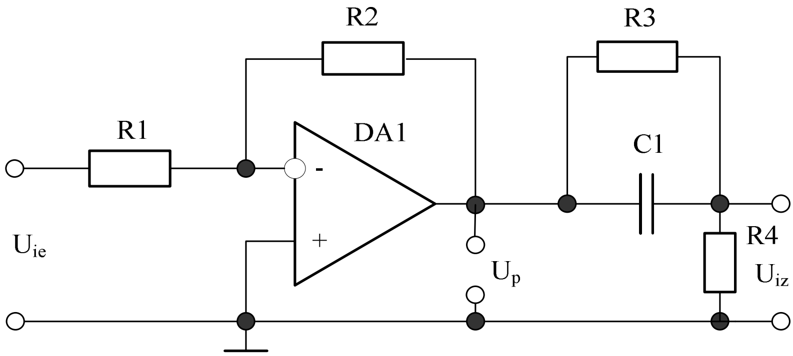
Attiecību $\mathbf{T_d/K_p = T_a}$ sauc par apsteidzes laiku. Jo lielāks apsteidzes laiks $\mathbf{T_a}$, jo efektīvāk darbojas diferencējošā ķēde, kas formē apsteidzošu iedarbi uz vadības objektu un paātrina tajā notiekošos procesus.

Pārvades funkcijas (4.21, 4.22) parāda, ka **PD - regulatoru** var realizēt divējādi:

- kā proporcionālā posma un diferencējošā posma **paralēlo** slēgumu;
- kā proporcionālā posma un diferencējošā filtra **virknes** slēgumu.

Vieglāk realizējams ir otrais variants, kur diferencējošais filtrs tiek izmantots kā virknes korekcijas iekārta, kas ļauj uzlabot AVS ātrdarbību un statisko precizitāti. **PD – regulatoru**, kuru apraksta pārvades funkcija (4.22), var izmantot inerciālu objektu, piemēram, siltuma ražošanas iekārtu vadīšanai.

Apskatīsim **PD – regulatora** ar **apsteidzošu iedarbi** praktisko realizāciju un modelēšanu. **PD – regulatora** principshēma sastāv no operacionālā pastiprinātāja **DA1** ar ieejas pretestību **R1** un aktīvu atgriezenisko saiti **R2**, kam virknē slēgts diferencējošais **CR** filtrs **(C1, R3, R4)** kā virknes korekcijas ķēde (4.4. att.).



4.4. att. **Elektroniskā PD–regulatora ar apsteidzošu iedarbi principshēma: DA1, R1, R2 – proporcionālais posms; C1, R3, R4 – diferencējošais filtrs**

<u>Piemērs</u>

Dots: operacionālā pastiprinātāja ieejas pretestība: **R1=500 Ω;** atgriezeniskās saites pretestība: **R2=1000 Ω;** diferencējošā filtra elementu parametri: **C_1=300 μF; R_3=150 kΩ; R_4=300 kΩ.**

Aprēķini: pastiprinātāja pārvades koeficients: **$K_p = U_p/U_{ie}$ = R2/R1 = 1000/500 = 2**;

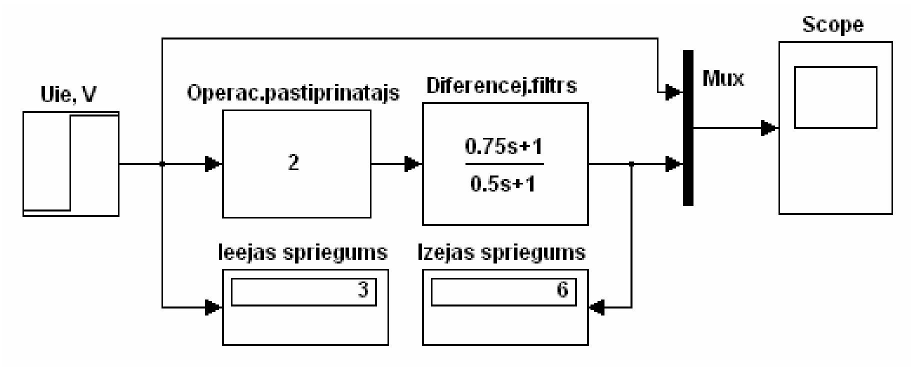
diferencējošā filtra laika konstante: $T_d = C_1 \cdot R_3 \cdot \left(\dfrac{R_4}{R_4 + R_3} \right) =$

$$= 300 \cdot 10^{-6} F \cdot 150 \cdot 10^3 \Omega \cdot \left(\frac{300}{300+150} \right) = 30s = 0.5\min.;$$

koriģētā laika konstante: **$T = T_d(1+1/K_p)$ = 0.5(1+1/2) = 0.75 min.**

4.4.1. PD – regulatora ar apsteidzošu iedarbi modelēšana

Sastādām modelēšanas blokshēmu (4.5. att.), kas sastāv no **„Simulink”** bibliotēkas blokiem: **„Step”** lēcienveida **ieejas sprieguma (U_{ie})** formēšanai; **„Slider Gain”** un **„Transfer Function”**, lai modelētu **operacionālo pastiprinātāju** un **diferencējošo filtru**; **„Display” – ieejas sprieguma** un nostabilizējušos **izejas sprieguma** nolasīšanai; **„Mux”** – raksturlīkņu attēla iegūšanai vienā koordinātu sistēmā uz osciloskopa **„Scope”** ekrāna.

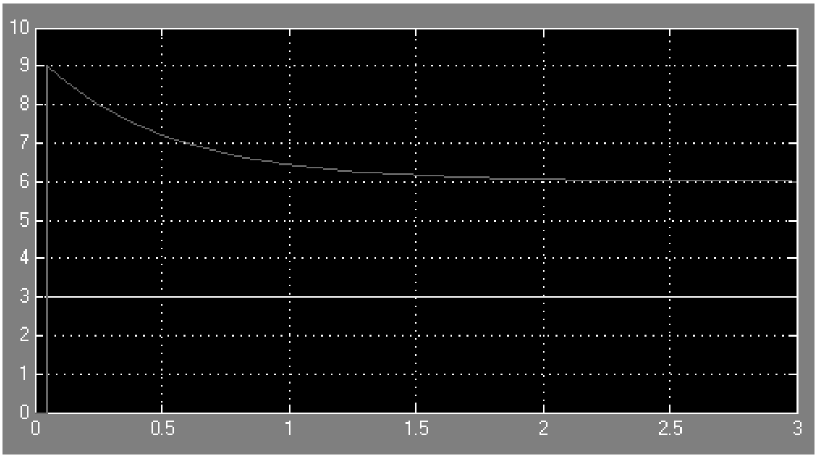


4.5. att. **PD – regulatora ar apsteidzošu iedarbi modelēšanas blokshēma**

Veicot simulāciju, iegūstam **PD – regulatora ar apsteidzošu iedarbi** pārejas procesa raksturlīkni **U_{iz} = f(t)** pie lēcienveida ieejas sprieguma izmaiņas: **U_{ie}= 0, t < 0.05 min; U_{ie} = 3V, t≥ 0.05 min** (4.6. att.).

Redzam, ka izejas sprieguma sākuma amplitūda **(U_{izmax} = 9V)** pārejas procesa beigās samazinās līdz statiskam lielumam **(U_{izs} = 6 V).** Šo aperiodiski rimstošo sākuma impulsu rada diferencējošā ķēde.

PD – regulatora izejas sprieguma sākuma un beigu lielumus var aprēķināt analītiski no pārvades funkcijas (4.22), izmantojot operatoru rēķinu teorēmas par funkcijas attēla sākuma un beigu lielumiem. Automātikas pamatos noskaidrojām sakarību starp reālā laika **t** un Laplasa argumenta **s** robežlielumiem: ja **t → 0, s** → ∞; ja **t** → ∞, **s → 0**.



4.6. att. **PD–regulatora ar apsteidzošu iedarbi pārejas procesa raksturlīknes**

Ievietojot argumenta **s** robežvērtības pārvades funkcijā, noteiksim **PD – regulatora** izejas sprieguma **sākuma** un **beigu** lielumus:

$$U_{iz_{\max}} = \lim_{s\to\infty} K_p \cdot \frac{T\cdot s+1}{T_d\cdot s+1}\cdot U_{ie} = K_p \cdot \frac{T}{T_d}\cdot U_{ie} = 2\cdot\frac{0.75}{0.5}\cdot 3 = 9V;$$

$$U_{izs} = \lim_{s\to 0} K_p \cdot \frac{T\cdot s+1}{T_d\cdot s+1}\cdot U_{ie} = K_p \cdot U_{ie} = 2\cdot 3 = 6V.$$

Aprēķinu rezultāti sakrīt ar modelēšanas rezultātiem. Apsteidzes sprieguma impulsa rimšanas laiku $\mathbf{t_r}$ nosaka laika konstante $\mathbf{T_d}$ **= 0.5 min.** $\mathbf{t_r}$ **≈ 4 ·**$\mathbf{T_d}$ **=3·0.5 = = 2min.**

4.4.2. PD–regulatora ar kavējošu iedarbi modelēšana

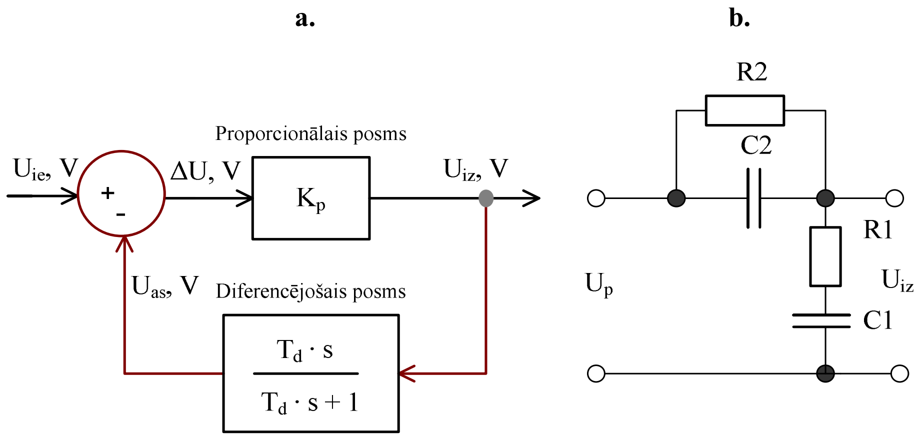
Ātrdarbīgās sistēmās, piemēram, dēļu gatera un daudzripzāģmašīnu vadības sistēmās izmanto **PD – regulatoru ar kavējošu iedarbi**, kas paaugstina sistēmas stabilitāti un samazina statisko kļūdu, jo ļauj palielināt proporcionālā posma pārvades koeficientu $\mathbf{K_p}$.

Šāds **PD – regulators** sastāv no proporcionālā posma, kuru aptver negatīva diferencējoša atgriezeniskā saite (4.7.a att.). Izmantojot atgriezeniskās saites slēguma īpašības, sastādām **PD – regulatora** pārvades funkciju:

$$W_{PD}(s)=\frac{U_{iz}(s)}{U_{ie}(s)}=\frac{K_p}{1+K_p\cdot\dfrac{T_d\cdot s}{T_d\cdot s+1}}=K_p\cdot\frac{T_d\cdot s+1}{T\cdot s+1}, \qquad (4.23)$$

kur laika konstante $\mathbf{T = (T_d + K_p \cdot T_d) = T_d \cdot (1 + K_p)}$.

Arī šajā gadījumā **PD–regulatoru** var realizēt ar operacionālā pastiprinātāja un diferencējošā filtra virknes slēgumu. Atšķirīga ir diferencējošā filtra shēma un parametri. Tā sastāv no kondensatoriem **C1, C2** un rezistoriem **R1, R2 (4.7.b. att.)**.



4.7. att. **Elektroniskais PD – regulators ar kavējošu iedarbi: a – algoritmiskā blokshēma; b – diferencējošā filtra principshēma**

Zinot kondensatoru kapacitātes un rezistoru pretestības, var aprēķināt filtra laika konstantes:

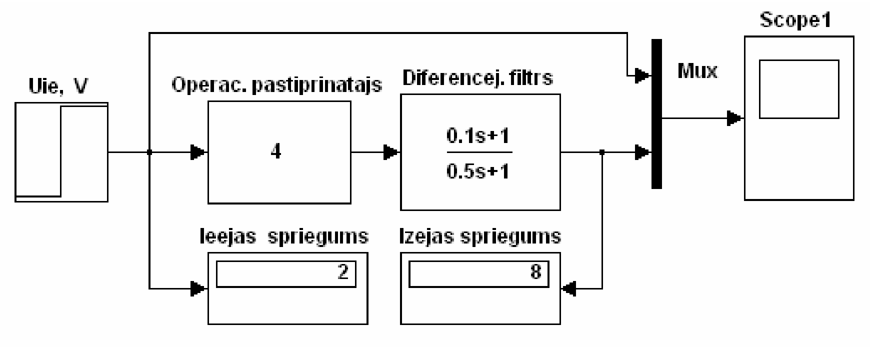
$$T_d = C_2 \cdot R_2\,; \quad T=\frac{R_1}{R_1+R_2}\cdot\left(\frac{C_1}{C_2}+1\right)T_d\,. \qquad (4.24)$$

Dots: operacionālā pastiprinātāja pārvades koeficients $\mathbf{K_p = 4}$;

diferencēšanas laika konstante $\mathbf{T_d = 0.1}$ **s**, filtra laika konstante **T = 0.5 s**.

Modelēšanas blokshēma un raksturlīknes`

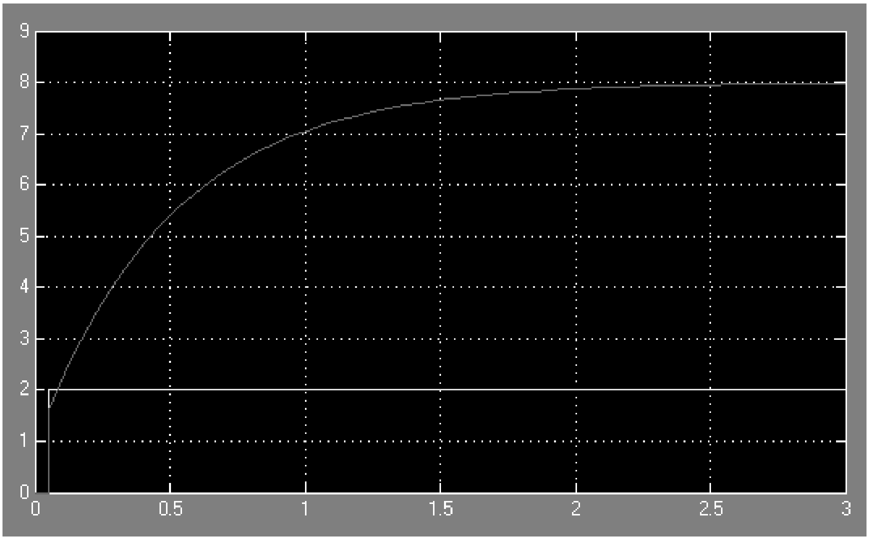
Sastādām modelēšanas blokshēmu (4.8. att.), kas atšķiras no iepriekšējās (4.5. att.) ar to, ka mainīti regulatora parametri. Ievadām $\mathbf{K_p = 4}$, $\mathbf{T_d = 0.1}$ **s** un **T = 0.5 s**. Ieejā padodam lēcienveida spriegumu **(U_{ie}= 0, ja t < 0.05 s; U_{ie} = 2 V, ja t ≥ 0.05 s).**



4.8. att. **PD – regulatora ar kavējošu iedarbi modelēšanas blokshēma**

Veicot simulāciju, iegūstam PD-regulatora ar kavējošu iedarbi izejas sprieguma pārejas procesa raksturlīkni $\mathbf{U_{iz} = f(t)}$, kas parāda, ka lēcienveida ieejas iedarbe tiek ierobežota (4.9. att.). Izejā ir neliels sprieguma lēciens $U_{iz0} < 2V$, kas pārejas procesa laikā pakāpeniski palielinās līdz stacionāram lielumam $U_{izs} = K_p \cdot U_{ie} = 4 \cdot 2 = 8V$.

Šāds **PD – regulators** nodrošina zāģmašīnas stabilu darbību pie izteikti svārstīgas slodzes, jo nolīdzina zāģmateriāla mainīgo parametru (zarainības, cietības, mitruma, ģeometrisko izmēru u.c.) ekstremālu iespaidu uz slodzes momenta strauju izmaiņu, kas savukārt pastiprina zāģu piedziņas elektrodzinēja strāvas svārstības un pasliktina zāģmašīnas AVS darbības stabilitāti un precizitāti.



4.9. att. **PD–regulatora ar kavējošu iedarbi pārejas procesa raksturlīknes**

Arī šajā gadījumā **PD – regulatora** izejas sprieguma sākuma un beigu lielumus var aprēķināt analītiski no pārvades funkcijas (4.23):

$$U_{iz\min} = \lim_{s\to\infty} K_p \cdot \frac{T_d \cdot s + 1}{T \cdot s + 1} \cdot U_{ie} = K_p \cdot \frac{T_d}{T} \cdot U_{ie} = 4 \cdot \frac{0.1}{0.5} \cdot 2 = 1.6V;$$

$$U_{izs} = \lim_{s\to 0} K_p \cdot \frac{T_d \cdot s + 1}{T \cdot s + 1} \cdot U_{ie} = K_p \cdot U_{ie} = 4 \cdot 2 = 8V.$$

Aprēķinu rezultāti sakrīt ar modelēšanas rezultātiem. Pie lēcienveida ieejas sprieguma izmaiņas, izejas spriegums aug pakāpeniski ar kavējumu, kura laiku $\mathbf{t_k}$ nosaka diferencējošā filtra laika konstante **T = 0.5 s.** Šo laiku var aprēķināt pēc aptuvenas formulas: $\mathbf{t_k} \approx$ **4·T = 4·0.5 = 2 s.**

4.5. Proporcionāli integrālā diferenciālā regulatora īpašības

Apvienojot proporcionālo, integrālo un diferenciālo vadības likumu iegūst PID vadības algoritmu, pēc kura darbojas **PID – regulators**.

PID – algoritma struktūru veido proporcionālā, integrējošā un diferencējošā posma paralēlais slēgums. PID – regulatora izejas signālu sastāda trīs komponentes. Tās mainās tieši proporcionāli ieejas signāla amplitūdai , tā izmaiņas ātrumam un integrālim no tā.

PID – regulatora dinamiskās īpašības novērtē pēc reakcijas uz lēcienveida ieejas iedarbi. Padodot lēcienveida signālu regulatora ieejā, vispirms sāk darboties diferencējošā ķēde, kas dod apsteidzošu izejas signāla komponenti. Diferencējošās ķēdes ģenerētais impulss ātri norimst. Tālāko procesa gaitu nosaka proporcionālā un integrējošā ķēde. Pēdējā pakāpeniski likvidē regulēšanas statisko kļūdu.

PID – regulators nodrošina visaugstāko procesa vadības kvalitāti, jo regulējošā iedarbe uz objektu tiek formētā atkarībā no trim faktoriem – ieejas signāla amplitūdas, tā izmaiņas ātruma un integrāļa. Tā kā **PID – regulatora** struktūru veido trīs paralēli savienoti posmi – proporcionālais, integrējošais un diferencējošais, tad tā darbības algoritms veidojas kā šo posmu algoritmu summa.

Idealizēta PID – regulatora dinamiskos pārejas procesus apraksta sekojošs **oriģinālvienādojums:**

$$X_{iz}(t) = K_p \cdot X_{ie}(t) + \frac{1}{T_i} \cdot \int_0^t X_{ie}(t)dt + T_d \cdot \frac{dX_{ie}(t)}{dt}, \qquad (4.25)$$

operatorvienādojums:

$$X_{iz}(s) = K_p \cdot X_{ie}(s) + \frac{X_{ie}(s)}{T_i \cdot s} + T_d \cdot X_{ie}(s) \cdot s \qquad (4.26)$$

un **pārvades funkcija:**

$$W_{PID} = \frac{X_{iz}(s)}{X_{ie}(s)} = K_p + \frac{1}{T_i \cdot s} + T_d \cdot s, \qquad (4.27)$$

kur $\mathbf{T_i}$ – integrējošā posma laika kostante, **s**;

$\mathbf{T_d}$ – difererencējošā posma laika konstante, **s**;

$\mathbf{K_p = X_{iz}/X_{ie}}$ – proporcionālā posma pārvades koeficients.

Parasti pārvades funkciju **(4.27)** pārveido, iznesot pirms iekavām $\mathbf{K_p}$:

$$W_{PID} = \frac{X_{iz}(s)}{X_{ie}(s)} = K_p(1 + \frac{1}{K_p \cdot T_i \cdot s} + \frac{T_d}{K_p} \cdot s) = K_p(1 + \frac{1}{T_{iz} \cdot s} + T_a \cdot s), \qquad (4.28)$$

kur $\mathbf{T_{iz} = K_p \cdot T_i}$ – izodroma laiks, **s**;

$\mathbf{T_a = T_d / K_p}$ - apsteidzes laiks, **s**;

PID – regulatora izodroma laiks ir $\mathbf{K_p}$ reizes lielāks par integrācijas laika konstanti $\mathbf{T_i}$, bet apsteidzes laiks ir $\mathbf{K_p}$ reizes mazāks par diferencēšanas laika konstanti $\mathbf{T_d}$. Mainot $\mathbf{K_p}$, vienlaicīgi jāmaina $\mathbf{T_{iz}}$ un $\mathbf{T_a}$.

Mūsdienu analogajos un ciparu kontrolleros instalētā **PID – regulatora** darbības algoritms ievērojami atšķiras no idealizētā, jo ideālu diferencējošu ķēdi tehniski nav iespējams realizēt. Tas jāņem vērā modelējot AVS ar **PID – regulatoru**. Izvēloties idealizēto algoritmu, modelēšanas rezultāts var būtiski atšķirties no reālā procesa.

Reālu PID – regulatoru visadekvātāk apraksta sekojoša pārvades funkcija:

$$W_{PIDr} = \frac{X_{iz}(s)}{X_{ie}(s)} = K_p \cdot (1 + \frac{1}{T_{iz} \cdot s}) \cdot (\frac{T_a \cdot s + 1}{T_f \cdot s + 1}), \qquad (4.29)$$

kur $\mathbf{T_f}$ – filtra laika konstante, **s**.

Dažādos avotos filtra laika konstante tiek uzdota plašā diapazonā $\mathbf{T_f = (0.05 \div \div 0.2)\ T_a}$. Lai diferencēšanas signāla amplitūda iekļautos reālās robežās, šo diapazonu ieteicams sašaurināt līdz $\mathbf{(0.1 \div 0.2)\ T_a}$.

Elektroniska PID – regulatora vienkāršota principshēma parādīta 4.10. attēlā. Shēma sastāv no operacionālā pastiprinātāja **DA1**, diferencējošās ķēdes rezistora **R1**

un kondensatora **C1**, integrējošās ķēdes atgriezeniskās saites kondensatora **C2** un rezistora **R2**.

Izmantojot izteiksmes (4.28 un 4.29) un veicot matemātiskus pārveidojumus, sastādīsim idealizēta un reāla elektroniska **PID – regulatora** pārvades funkcijas.

Idealizēta **elektroniska PID – regulatora** pārvades funkcija:

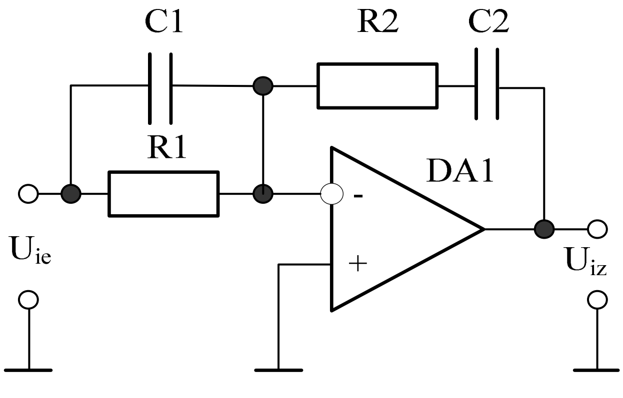
$$W_{PIDi} = \frac{U_{iz}(s)}{U_{ie}(s)} = K_p \cdot (1 + \frac{1}{T_{iz} \cdot s} + T_a \cdot s) = K_p \cdot (\frac{T_{iz} \cdot s + 1}{T_{iz} \cdot s} + T_a \cdot s). \qquad (4.30)$$

Reāla **elektroniska PID – regulatora** pārvades funkcija:

$$W_{PIDr} = \frac{U_{iz}(s)}{U_{ie}(s)} = K_p \cdot (1 + \frac{1}{T_{iz} \cdot s}) \cdot (\frac{T_a \cdot s + 1}{T_f \cdot s + 1}) = K_p \cdot \frac{T_{iz} \cdot s + 1}{T_{iz} \cdot s} \cdot \frac{T_a \cdot s + 1}{T_f \cdot s + 1}. \qquad (4.31)$$

Idealizētā un reālā modeļa koeficientus var izteikt ar elektroniskās ķēdes parametriem:

- pārvades koeficients $\mathbf{K_p = R_2 / R_1}$;
- diferencēšanas laika konstante $\mathbf{T_d = C_1 \cdot R_2}$;
- apsteidzes laiks $\mathbf{T_a = T_d / K_p = (C_1 \cdot R_2) / (R_2 / R_1) = C_1 \cdot R_1}$;
- integrēšanas laika konstante $\mathbf{T_i = C_2 \cdot R_1}$;
- izodroma laiks $\mathbf{T_{iz} = T_i \cdot K_p = C_2 \cdot R_1 \cdot R_2 / R_1 = C_2 \cdot R_2}$.



4.10. att. **Elektroniskā PID – regulatora vienkāršota principshēma**

4.5.1. Idealizēta PID – regulatora modelēšana

Modelēšanas blokshēma (4.11. att.), sastāv no **„Simulink”** blokiem: **„Slider Gain”(Kp)** – proporcionālā posma; **„Transfer Function” (Wi)** – integrējošā posma; **„Slider Gain”(Td)** un **„Derivative”** – diferencējošā posma, kuri attiecīgi modelē **PID – regulatora** proporcionālo ķēdi, integrējošo ķēdi un diferencējošo ķēdi. Lēcienveida spriegumu regulatora ieejā formē ar bloku **„Step” (Uie).** Ciparu displeji (**Ieejas spriegums, Up** un **Izejas spriegums**) uzstādīti, lai nolasītu regulatora ieejas spriegumu, proporcionālā posma izejas spriegumu un regulatora izejas spriegumu.

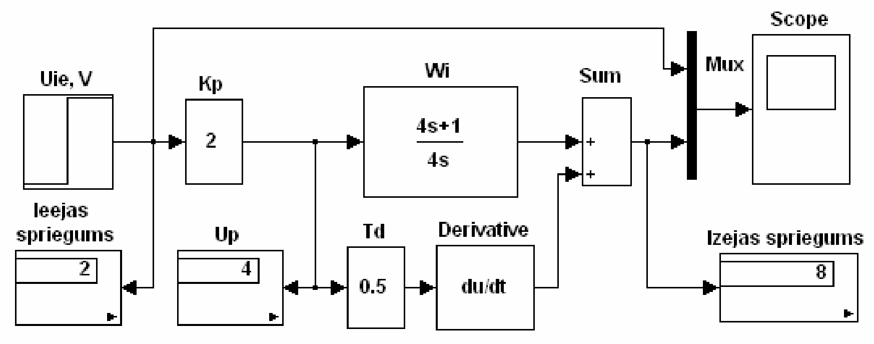
Dots: proporcionālās ķēdes pārvades koeficients $\mathbf{K_p = 2}$;

integrējošās ķēdes laika konstante $\mathbf{T_i = 2\ s}$;

izodroma laiks $\mathbf{T_{iz} = T_i \cdot K_p = 2 \cdot 2 = 4\ s}$;

diferencējošās ķēdes laika konstante $\mathbf{T_d = 1\ s}$;

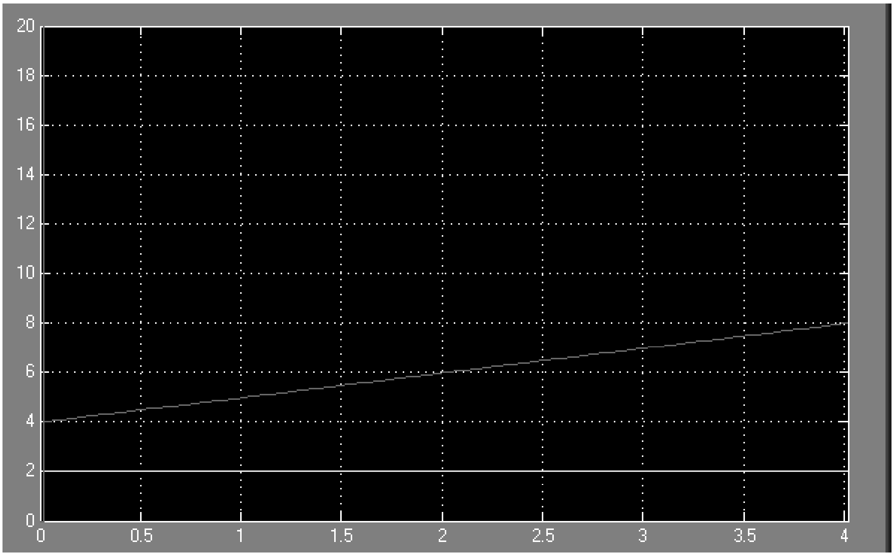
apsteidzes laiks $\mathbf{T_a = T_d / K_p = 1/2 = 0.5\ s}$.



4.11. att. **Idealizēta PID – regulatora pārejas procesu modelēšanas blokshēma**

Veicot simulāciju, iegūstam **idealizēta PID – regulatora** pārejas procesa raksturlīknī $\mathbf{U_{iz} = f(t)}$ pie lēcienveida ieejas sprieguma izmaiņas $\mathbf{(U_{ie} = 0,\ t < 0.02\ s;}$ $\mathbf{U_{ie} = 2\ V,\ t \geq 0.02\ s)}$ (4.12. att.). Pārejas procesa sākuma laiks izvēlēts lielāks par nulli, lai būtu labi redzama sprieguma lēcienveida izmaiņa. Varam pārliecināties, ka izejas sprieguma sākuma amplitūda ir neierobežoti liela $\mathbf{(U_{izmax} \to \infty)}$, kas momentāni sabrūk līdz statiskam lielumam $\mathbf{U_p = 4\ V}$. Darbojoties integrējošajai ķēdei, šis spriegums pakāpeniski paaugstinās (4.12. att.).

Idealizētā PID – regulatora izejas sprieguma sākuma lielumu $\mathbf{U_{iz0}}$ **(t = 0)** un beigu lielumu $\mathbf{U_{izb}}$ brīvi izvēlēta laika perioda beigās ,piemēram, $\mathbf{t_b = 4\ s}$, pie nulles sākuma nosacījumiem var aprēķināt analītiski no pārvades funkcijas (4.30). Izmantojot operatoru rēķinu sakarības **(ja t → 0, s → ∞; ja t → $\mathbf{t_b}$, s → 1 / $\mathbf{t_b}$)** un ņemot vērā, ka idealizēta PID – regulatora diferencējošās ķēdes darbības laiks faktiski vienāds ar nulli (sk. 4.12. att.), pārvades funkcijā (4.30) var ievietot $\mathbf{T_a = 0}$.



4.12. att. **Idealizēta PID – regulatora pārejas procesa raksturlīknes $X_{ie}(t)$, $X_{iz}(t)$**

Ievietojot skaitliskos lielumus $\mathbf{K_p=2}$, $\mathbf{T_{iz}=4s}$ un $\mathbf{s=1/t_b=1/4=0.25\ s^{-1}}$, aprēķinām sākuma un beigu spriegumus:

$$U_{iz0(t=0)} = \lim_{s\to\infty} K_p \cdot \frac{T_{iz}\cdot s + 1}{T_{iz}\cdot s} \cdot U_{ie} = 2\cdot\infty\cdot 2 = \infty;$$

$$U_{izb(t_b=4s)} = \lim_{s\to 0.25} K_p \cdot \frac{T_{iz}\cdot s + 1}{T_{iz}\cdot s} \cdot U_{ie} = 2\cdot\frac{4\cdot 0.25 + 1}{4\cdot 0.25}\cdot 2 = 8V.$$

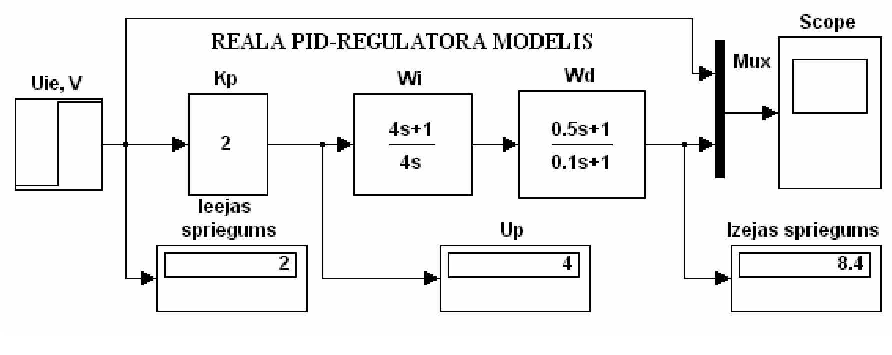
Aprēķinu rezultāti precīzi sakrīt ar modelēšanas rezultātiem (4.12. att.). Ar idealizēto modeli iegūst neadekvātu rezultātu. Reāls **PID – regulators** nevar ģenerēt sprieguma impulsu ar bezgalīgi lielu amplitūdu un ar bezgalīgi stāvu fronti.

4.5.2. Reāla PID – regulatora modelēšana

Modelēšanas blokshēma sastādīta atbilstoši pārvades funkcijai (4.31). **Reāla PID – regulatora** modelis sastāv no trim virknē slēgtiem **„Simulink"** blokiem: **„Slider Gain" (Kp)** – proporcionālā posma; **„Transfer Function" (Wi)** – integrējošā posma; **„Transfer Function" (Wd)** – diferencējošā posma, kuri attiecīgi modelē **PID – regulatora** proporcionālo, integrējošo un diferencējošo ķēdi. Diferencējošo ķēdi veido diferencējošais filtrs. Lēcienveida spriegumu regulatora ieejā formē ar bloku **„Step"** (**Uie**) (4.13. att.).

Lai korekti salīdzinātu reālo modeli ar idealizēto modeli, izvēlamies vienādus to parametrus: pārvades koficientu $\mathbf{K_p = 2}$; izodroma laiku $\mathbf{T_{iz} = 4}$ **s** un apsteidzes laiku $\mathbf{T_a = 0.5}$ **s**.

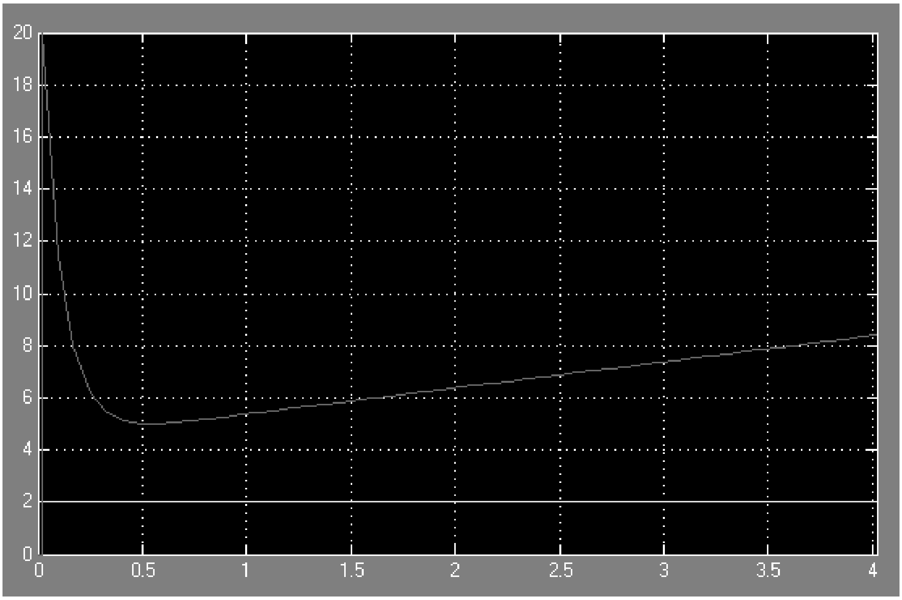
Reālajā modelī papildus jāievada diferencēšānas filtra laika konstante, ko izvēlamies no nosacījuma: $\mathbf{T_f = 0.2 \cdot T_a = 0.2 \cdot 0.5 = 0.1}$ **s.**



4.13. att. **Reāla PID – regulatora pārejas procesu modelēšanas blokshēma**

Veicot simulāciju, iegūstam **reāla PID – regulatora** pārejas procesa raksturlīkni $\mathbf{U_{iz} = f(t)}$ pie lēcienveida ieejas sprieguma izmaiņas **(U_{ie} = 0, t < 0.02 s; U_{ie} = 2 V, t ≥ 0.02 s)** (4.14. att.).

Izejas sprieguma sākuma impulsa amplitūda ir ierobežota **(U_{iz0} = 20 V)**. Tā norimst aperiodiski ar laika konstanti **T_f = 0.1 s**. Impulsa laikā vienlaicīgi darbojas diferencējošā, integrējošā un proporcionālā ķēde. Pēc tam, kad beidzies diferencējošās ķēdes pārejas process, integrējošā ķēde pakāpeniski paaugstina regulatora izejas spriegumu un likvidē regulēšanas statisko kļūdu.



4.14. att. **Reāla PID–regulatora pārejas procesa raksturlīknes $X_{ie}(t)$, $X_{iz}(t)$**

Pārejas procesa sākumā, kad darbojas diferencējošā ķēde, būtiski atšķiras reālā modeļa izejas spriegums un tā izmaiņas raksturs salīdzinājumā ar idealizēto modeli (sk. 4.12. att. un 4.14. att.). Proporcionālās un integrējošās ķēdes darbību abi modeļi apraksta vienādi. Modelēšanas laika beigās abu modeļu izejas spriegums atšķiras par lielumu: γ_{Uiz} **= (8.4-8)/8.4·100 % ≈ 5 %.** Modeļu atšķirības visvairāk iespaido pārejas procesa sākumu. Taču arī beigu rezultāta kļūda 5% ir vērā ņemama.

Reālā PID – regulatora izejas sprieguma sākuma lielumu $\mathbf{U_{iz0}}$ **(t = 0)** un beigu lielumu $\mathbf{U_{izb}}$ modelēšanas laika perioda beigās $\mathbf{t_b}$ **= 4 s**, pie nulles sākuma nosacījumiem, var aprēķināt analītiski no pārvades funkcijas (4.31), izmantojot operatoru rēķinu sakarības **(ja t → 0, s → ∞; ja t → $\mathbf{t_b}$, s → 1/ $\mathbf{t_b}$):**

$$U_{iz0(t=0)} = \lim_{s\to\infty} K_p \cdot \frac{T_{iz}\cdot s+1}{T_{iz}\cdot s}\cdot\frac{T_a\cdot s+1}{T_f\cdot s+1}\cdot U_{ie} = K_p\cdot\frac{T_a}{T_f}\cdot U_{ie} = 2\cdot\frac{0.5}{0.1}\cdot 2 = 20V;$$

$$U_{izb(t_b=4s)} = \lim_{s\to 0.25} K_p \cdot \frac{T_{iz}\cdot s+1}{T_{iz}\cdot s}\cdot\frac{T_a\cdot s+1}{T_f\cdot s+1}\cdot U_{ie} = 2\cdot\frac{4\cdot 0.25+1}{4\cdot 0.25}\cdot\frac{0.5\cdot 0.25+1}{0.1\cdot 0.25+1}\cdot 2 = 8.78V.$$

Modelēšanas laika beigās **($\mathbf{t_b}$ = 4 s)** aprēķinātais izejas spriegums (**8.78 V**) atšķiras no modelēšanas rezultāta (**8.4 V**) par lielumu: $\boldsymbol{\gamma_{Uizb}}$**=(8.78-8.4)/8.4·100%≈5%.** Tas izskaidrojams ar regulēšanas statisko kļūdu, kas aprēķinā netiek ņemta vērā. Varam pārliecināties, ka, augot laikam, statisko kļūdu pakāpeniski samazina integrējošā ķēde. Pie t_b=10 sekundes modelētā un aprēķinātā lieluma atšķirības nepārsniedz 2 %.

Izmantojot PID – regulatoru ar atbilstoši iestatītiem un ar vadības objektu saskaņotiem parametriem, var iegūt augstu procesa vadības kvalitāti

4.6. Regulatora izvēle statiskiem objektiem ar transportkavējumu

Vairāk kā 80 % tehnoloģisko sistēmu dinamiskos procesus apraksta otrās un augstāku kārtu inerciālie statiskie posmi, kurus aproksimācijas ceļā var aizstāt ar pirmās kārtas inerciāla posma un kavējumposma virknes slēgumu. Tas dod iespēju unificēt daudzu tehnoloģisko iekārtu darbības algoritmus un izstrādāt vienotus nosacījumus to vadības algoritmu izvēlei.

4.6.1. Lernera kritēriji un vadības algoritma izvēles nosacījumi

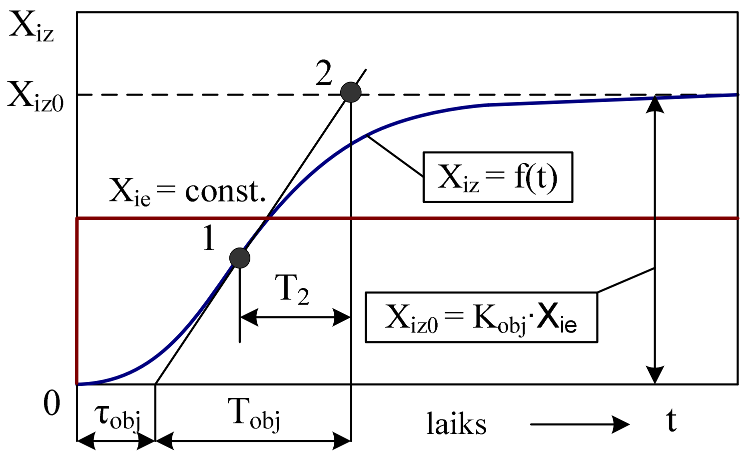
Apskatīsim tehnoloģisko objektu, piemēram, tvaika katlu, kā otrās kārtas inerciālu statisku posmu, kuru apraksta sekojoša pārvades funkcija:

$$W = \frac{K_{obj}}{T_1^2 s^2 + T_2 s + 1}, \tag{4.32}$$

kur $\mathbf{K_{obj}}$ – vadības objekta pārvades koeficients;

$\mathbf{T_1}$un $\mathbf{T_2}$ - vadības objekta laika konstantes, **s**.

Pieņemsim, ka vadības objekta pārejas procesa raksturlīkne $\mathbf{X_{iz} = f(t)}$ uzņemta eksperimentāli, padodot ieejā konstantu lēcienveida iedarbi $\mathbf{X_{ie}}$ = **const** (4.15. att.).



4.15. att. **Otrās kārtas inerciāla statiska tehnoloģiskā objekta pārejas procesa raksturlīknes: $\mathbf{X_{ie} = f(t)}$ un $\mathbf{X_{iz} = f(t)}$ ($\mathbf{t < 0, X_{ie} = 0; t \geq 0; X_{ie} = const.}$)**

Raksturlīknei $\mathbf{X_{iz} = f(t)}$ ir pārliekuma punkts 1. Lai grafoanalītiski noteiktu laika konstantes $\mathbf{T_1}$ un $\mathbf{T_2}$, velkam šajā punktā pieskari. Atzīmējam pieskares krustpunktu **2** ar ordināti $\mathbf{X_{iz0}}$**.** Uz laika ass **t** atliekam trīs nogriežņus, kas apzīmēti ar $\boldsymbol{\tau_{obj}}$, $\mathbf{T_{obj}}$ un $\mathbf{T_2}$. Pēdējā ir meklētā laika konstante $\mathbf{T_2}$. Laika konstanti $\mathbf{T_1}$ aprēķina pēc sekojošas formulas: $T_1 = \sqrt{\tau_{obj}\left(T_2 - \tau_{obj}\right)}$.

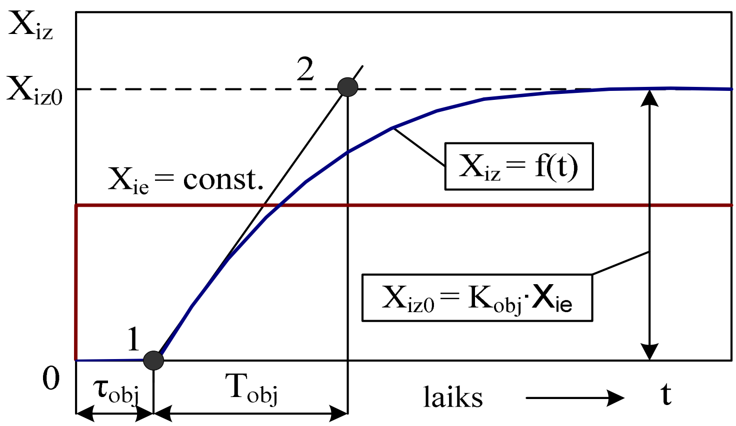
Iegūtās laika konstantes $\boldsymbol{\tau_{obj}}$ un $\mathbf{T_{obj}}$ dod iespēju otrās kārtas inerciālu statisku posmu aproksimēt kā pirmās kārtas statisku posmu ar transportkavējumu (4.16. att.).

Aproksimētā posma pārvades funkciju izsaka sekojoši:

$$W_{obj}(s) = \frac{K_{obj} \cdot e^{-s \cdot \tau_{obj}}}{T_{obj} \cdot s + 1}, \qquad (4.33)$$

kur $\boldsymbol{\tau_{obj}}$ – vadības objekta transportkavējuma laiks, **s**;

$\mathbf{T_{obj}}$ – vadības objekta laika konstante, **s**.



4.16. att. **Otrās kārtas inerciālā statiskā posma aproksimētā pārejas procesa raksturlīkne: X_{iz}= f(t) (t < 0, X_{ie}= 0; t ≥ 0; X_{ie}= const.)**

Veiktā aproksimācija dod iespēju izmantot **Lernera** un **Cīglera – Nikolsa** kritējus, lai izvēlētos tehnoloģiskā objekta vadības kontrollera darbības algoritmu un optimāli iestatītu tā parametrus. Minētie kritēriji sastādīti tehnoloģiskajiem objektiem, kurus apraksta pirmās kārtas aperiodisks posms ar transportkavējumu. Transportkavējums sarežģī tehnoloģiskā procesa vadību. Jo lielāks objekta transporkavējuma laiks $\boldsymbol{\tau_{obj}}$ attiecībā pret tā laika konstanti $\mathbf{T_{obj}}$, jo lielākas problēmas rada procesa vadības kvalitātes nodrošināšana. Tāpēc lielākiem $\boldsymbol{\tau_{obj}}$ jāizvēlas augstākas klases kontrolleris ar sarežģītāku vadības algoritmu (4.1. tab.).

4.1.tabula

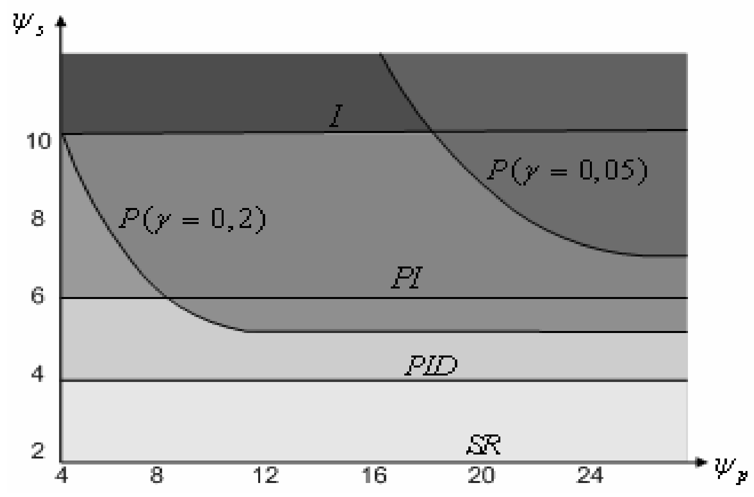
Vadības algoritma izvēles kritēriji statiskiem objektiem ar transportkavējumu

$\tau_{obj}/T_{obj} \leq 0{,}1$	Izmantojami diskrētās vadības kontrolleri ar divpozīciju algoritmu
$0.1 \leq \tau_{obj}/T_{obj} \leq 0.25$	Izmantojami analogās vadības kontrolleri ar PI vai PID algoritmu
$0.25 \leq \tau_{obj}/T_{obj} \leq 0.5$	Izmantojami diskrētās vadības kontrolleri ar impulsregulēšanas algoritmu

Pirmās kārtas statiska objekta ar transportkavējumu vadības algoritma izvēlei var izmantot Lernera diagrammu (4.17. att.). Uz diagrammas asīm atlikti relatīvie laiki. Uz vertikālo asi atlikta laika konstantes $\mathbf{T_{obj}}$ attiecība pret transportkavējumu $\boldsymbol{\tau_{obj}}$, uz horizontālo asi atlikta pārejas procesa laika $\mathbf{t_p}$ attiecība pret transportkavējumu:

$$\psi_s = T_{obj} / \tau_{obj}; \; \psi_p = t_p / \tau_{obj} .$$

Horizontālās līnijas nodala vadības algoritmu pielietošanas apgabalus.



4.17. att. **Lernera diagramma statiska vadības objekta ar transportkavējumu vadības algoritma izvēlei** (I – integrālais regulators; P – proporcionālais regulators; PI – proporcionāli integrālais regulators; PID – proporcionāli integrālais diferenciālais regulators; SR – speciālais impulsregulators)

P – regulatora pielietošanas apgabalu nosaka stabilizējamā tehnoloģiskā parametra statiskā lieluma $\mathbf{X_{izs}}$ pieļaujamā relatīvā novirze γ no uzdotā lieluma $\mathbf{X_{iz0}}$, kur $\gamma = \left| (\mathbf{X_{iz0}} - \mathbf{X_{izs}}) / \mathbf{X_{iz0}} \right|$. Jo lielāks pieļaujamais γ, jo plašāks **P – regulatora** pielietošanas apgabals (4.17. att.). Automātiskās regulēšanas algoritmu izvēles rekomendācijas atbilstoši Lernera kritērijiem dotas 4.2.tabulā.

4.2.tabula

Regulēšanas algoritma izvēles nosacījumi

$\psi_s = T_{obj} / \tau_{obj}$; $\psi_p = t_p / \tau_{obj}$, kur $\mathbf{t_p}$ – pārejas procesa laiks, **s**; T_{obj} – vadības objekta laika konstante, s; τ_{obj} - transportkavējuma laiks, s	
$2 < \psi_s < 4$	Izvēlas impulsregulēšanas algoritmus
$4 < \psi_s < 6$	Ieteicama PID vadība
$6 < \psi_s < 10$	Var izvēlēties P, PI vai PID vadību
$\psi_s > 10$	Var izvēlēties divpozīciju regulēšanu

4.6.2. Vadības algoritma parametru iestatīšanas kritēriji

20. gs. 40-tajos gados **Cīglers** (Ziegler) un **Nikolss** (Nichols) izstrādāja kritērijus analogo regulatoru parametru optimālai iestatīšanai, kas galvenokārt bija paredzēti statisku tehnoloģisko objektu ar transportkavējumu vadības realizācijai. Kritēriju pamatā ir vadības objekta reakcija uz lēcienveida iedarbi (ieejas vai perturbācijas). Balstoties uz vadības objekta parametriem un tā izejas lieluma izmaiņas stāvumu – $\mathbf{S = K_{obj} \cdot X_{ie} / T_{obj}}$, nosaka vadības iekārtas optimālos iestatījumus.

Pamatojoties uz šiem praksē pārbaudītajiem kritērijiem, sastādītas matemātiskas izteiksmes (4.3. tab.), kas dod iespēju aprēķināt dažāda tipa analogo regulatoru ar proporcionālo **(P)**, integrālo **(I)**, proporcionāli integrālo **(PI)** un proporcionāli integrālo diferenciālo **(PID)** vadības algoritmu parametru iestatījumus atkarībā no statiska inerciāla vadības objekta parametriem: statiskā pārvades koeficienta $\mathbf{K_{obj}}$, transportkavējuma laika $\boldsymbol{\tau_{obj}}$ un laika konstantes $\mathbf{T_{obj}}$.

Vispirms pēc Lernera diagrammas izvēlas vadības algoritmu (3.17. att.). Pēc tam izvēlas pārejas procesa veidu:

- monotonu (aperiodisku), kura norise notiek bez pārregulējuma;
- ar maksimālo pārregulējumu, kas nepārsniedz **20 %**;
- pēc minimālā integrālā kvalitātes kritērija $\mathbf{I_{min} = f(\varepsilon (t))}$, kur **ε (t)** ir regulēšanas relatīvā dinamiskā kļūda.

Pēc pārejas procesa veida izvēles veic automātiskās vadības iekārtas parametru iestatījumu aprēķinu, izmantojot 4.3. tabulā dotos kritērijus.

4.3.tabula

Vadības algoritmu parametru iestatīšanas kritēriji un aprēķina formulas

Regulatora tips	**Pārejas procesa veids**		
	aperiodisks	**ar 20% pārregulējumu**	**ar minimālu integrālo kritēriju** $I_{\min} = \int_0^\infty \varepsilon^2(t)dt$
P	$K_p = \frac{0.3 \cdot T_{obj}}{K_{obj}\tau_{obj}}$	$K_p = \frac{0.7 \cdot T_{obj}}{K_{obj}\tau_{obj}}$	$K_p = \frac{0.9 \cdot T_{obj}}{K_{obj}\tau_{obj}}$
I	$k_i = \frac{1}{4.5 K_{obj} T_{obj}}$	$k_i = \frac{1}{1.7 K_{obj} T_{obj}}$	$k_i = \frac{1}{1.7 K_{obj} T_{obj}}$
PI	$K_p = \frac{0.6 \cdot T_{obj}}{K_{obj}\tau_{obj}}$, $T_{iz} = 0.6 T_{obj}$	$K_p = \frac{0.7 \cdot T_{obj}}{K_{obj}\tau_{obj}}$, $T_{iz} = 0.7 T_{obj}$	$K_p = \frac{1 \cdot T_{obj}}{K_{obj}\tau_{obj}}, T_{iz} = T_{obj}$
PID	$K_p = \frac{0.95 \cdot T_{obj}}{K_{obj}\tau_{obj}},$ $T_{iz} = 2.4\tau,\ T_a = 0.4\tau$	$K_p = \frac{1.2 \cdot T_{obj}}{K_{obj}\tau_{obj}},$ $T_{iz} = 2\tau,\ T_a = 0.4\tau$	$K_p = \frac{1.4 \cdot T_{obj}}{K_{obj}\tau_{obj}},$ $T_{iz} = 1.3\tau,\ T_a = 0.5\tau$

Augstāku kārtu **inerciāliem statiskiem objektiem kā optimālu parasti izvēlas pārejas procesu ar vienu pārregulējumu**, kurš mazāks par **20 %**. Ja pārregulējums nepārsniedz **10 %**, tad pārejas procesa kvalitāte parasti ir atbilstoša kā no stabilitātes, tā ātrdarbības viedokļa.

Mazinerciāliem vadības objektiem – ātrdarbīgās elektropiedziņas un sekošanas sistēmās – svarīgs kvalitātes rādītājs ir pārejas procesa norises (regulēšanas) laiks $\mathbf{t_p}$. Tad jāmeklē optimāls kompromiss starp vadības sistēmas **stabilitāti un ātrdarbību**.

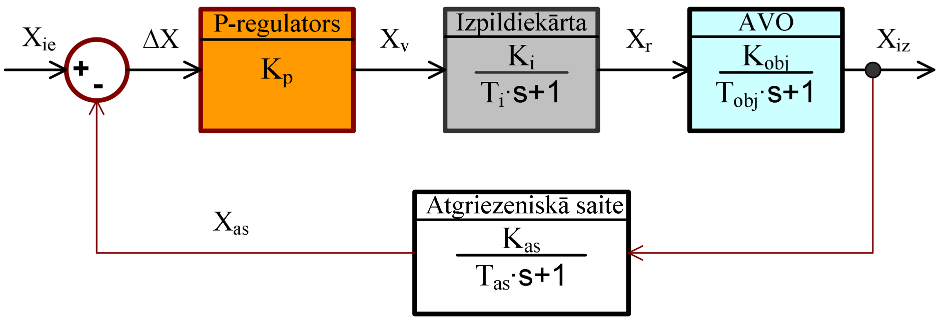
Ņemot vērā to, ka tehnoloģiskā objekta **matemātiskais modelis nav ideāls**, arī aprēķinātie vadības algoritma iestatījumi nebūs optimāli. Tādēļ tie ir atkārtoti **jāprecizē modelēšanas procesā**, vadoties pēc pārejas procesa kvalitātes rādītājiem. Šajā nozīmē **lielas priekšrocības ir imitāciju modelēšanai Windows vidē**.

5. AUTOMĀTISKO SISTĒMU MODELĒŠANAS PIEMĒRI

5.1. Slēgtas AVS darbības analīze ar frekvenču metodi

Kā piemēru apskatīsim lineāru AVS ar inerciālu **negatīvu atgriezenisko saiti**, kas sastāv no **proporcionālā regulatora** ar pārvades koeficientu $\mathbf{K_p}$, inerciālas **izpildiekārtas** ar pārvades koeficientu $\mathbf{K_i}$ un laika konstanti $\mathbf{T_i}$, statiska automātiskās **vadības objekta (AVO)** ar pārvades koeficientu $\mathbf{K_{obj}}$ un laika konstanti $\mathbf{T_{obj}}$.

Atgriezeniskajā saitē slēgts inerciāls **mērīšanas pārveidotājs** ar pārvades koeficientu $\mathbf{K_{as}}$ un laika konstanti $\mathbf{T_{as}}$ (5.1. att.). Diferenciālā ieejas shēma formē novirzes signālu $\Delta \mathbf{X} = \mathbf{X_{ie}} - \mathbf{X_{as}}$.



5.1. att. **Slēgtas AVS ar inerciālu atgriezenisko saiti algoritmiskā blokshēma**

<u>**Vaļējas un slēgtas AVS pārvades funkcijas**</u>

Lai iegūtu slēgtas AVS pārvades funkciju, vispirms sastāda vaļējas sistēmas pārvades funkciju $\mathbf{W_v}$. Šai nolūkā pārtrauc atgriezenisko saiti. Tad iegūst vaļēju AVS, kas sastāv no virknē slēgtiem blokiem: **P – regulatora**, **Izpildiekārtas** un automātiskās vadības objekta **AVO**.

Izmantojot AVS posmu virknes slēguma īpašības, sastādām **vaļējas sistēmas pārvades funkciju:**

$$W(s)=\frac{X_{iz}(s)}{\Delta X(s)}=\frac{K_p\cdot K_i\cdot K_{obj}}{(T_i\cdot s+1)(T_{obj}\cdot s+1)}=\frac{K}{(T_i\cdot s+1)(T_{obj}\cdot s+1)},\qquad(5.1)$$

kur **K = $K_p \cdot K_i \cdot K_{obj}$** – vaļējas sistēmas pārvades koeficients;

ΔX(s) – novirzes signāla attēls;

X_{iz}(s) – izejas signāla attēls.

Veicot nelielus algebriskus pārveidojumus, funkciju (5.1) uzrakstām sekojošā formā:

$$W(s)=\frac{K}{T_i\cdot T_{obj}\cdot s^2+(T_i+T_{obj})\cdot s+1}=\frac{K}{a_0\cdot s^2+a_1\cdot s+1},\qquad(5.2)$$

kur **$a_0 = T_i \cdot T_{obj}$, $a_1 = (T_i + T_{obj})$** – konstanti koeficienti ar laika mērvienību;

Zinot vaļējas AVS pārvades funkciju un izmantojot atgriezeniskās saites slēguma īpašības, sastādām **slēgtas AVS pārvades funkciju**:

$$\Phi=\frac{X_{iz}(s)}{X_{ie}(s)}=\frac{W_v}{1+W_v\cdot W_{as}}=\frac{1}{\frac{1}{W_v}+W_{as}},\qquad(5.3)$$

kur **$W_{as} = K_{as} / (T_{as} \cdot s+1)$** – atgriezeniskās saites pārvades funkcija.

Ievietojot formulā (5.2) funkciju **W_v** un **W_{as}** izteiksmes un izdarot matemātiskus pārveidojumus, iegūstam **slēgtas AVS pārvades funkciju** izvērstā veidā:

$$\Phi=\frac{K\cdot(T_{as}\cdot s+1)}{(T_i\cdot s+1)(T_{obj}\cdot s+1)(T_{as}\cdot s+1)+K_{as}\cdot K}.\qquad(5.4)$$

Izdarot matemātiskus pārveidojumus un ievedot jaunus apzīmējumus, iegūstam sekojošu izteiksmi:

$$\Phi=\frac{b_0\cdot s+b_1}{a_0\cdot s^3+a_1\cdot s^2+a_2\cdot s+a_3}=\frac{Q(s)}{D(s)},\qquad(5.5)$$

kur $\mathbf{Q(s) = b_0 \cdot s + b_1}$ – slēgtas sistēmas iedarbes operators;

$\mathbf{D(s) = a_0 \cdot s^3 + a_1 \cdot s^2 + a_2 \cdot s + a_3}$ - slēgtas sistēmas pašoperators;

$\mathbf{b_0 = K \cdot T_{as},\ b_1 = K,\ a_0 = T_i \cdot T_{obj} \cdot T_{as}\ ,\ a_1 = T_i \cdot T_{obj} + T_i \cdot T_{as} + T_{obj} \cdot T_{as},}$

$\mathbf{a_2 = T_i + T_{obj} + T_{as},\ a_3 = 1 + K_{as} \cdot K}$ – konstanti koeficienti.

Pielīdzinot nullei iedarbes operatoru **Q(s)**, iegūst vienādojumu, kura saknes sauc par **„nullēm”**.

Dotajai sistēmai ir viena **„nulle”** $\mathbf{(b_0 \cdot s + b_1 = 0,\ s_0 = -\ b_1 / b_0 = -\ 1 / T_{as})}$.

Pielīdzinot nullei slēgtas sistēmas pašoperatoru **D(s)**, iegūst tās raksturīgo vienādojumu, kura saknes sauc par **„poliem”.** Dotajai sistēmai ir trīs **„poli”-** $\mathbf{s_1}$**,** $\mathbf{s_2}$ un $\mathbf{s_3}$**,** kurus iegūst atrisinot algebrisku vienādojumu: $\mathbf{a_0 \cdot s^3 + a_1 \cdot s^2 + a_2 \cdot s + a_3 = 0}$**.**

Veiktā analīze ļauj izdarīt secinājumu, ka aptverot otrās kārtas vaļēju AVS ar negatīvu inerciālu atgriezenisko saiti iegūst trešās kārtas slēgtu AVS. Tātad inerciāla atgriezeniskā saite izmaina slēgtas AVS dinamiskās īpašības.

Vaļējas AVS frekvenču funkcija

Automātikas pamatos noskaidrojām, ka AVS stabilitātes un darbības kvalitātes analīzei var izmantot **frekvenču funkcijas**. Ja AVS ieejā padod harmonisku signālu, tad izejā arī iegūst harmonisku signālu ar **tādu pašu frekvenci**, bet ar **atšķirīgu amplitūdu un fāzi**, kuras atkarīgas no **AVS parametriem** un **ieejas signāla frekvences**. Palielinot ieejas signāla frekvenci, samazinās izejas signāla amplitūda, bet aizkavēšanās jeb nobīde fāzē attiecībā pret ieejas signālu palielinās.

Ja AVS izejā rodas nerimstošas svārstības, tad tās raksturīgā vienādojuma saknes ir imagināras. Tādēļ, lai iegūtu AVS frekvenču funkciju, tās pārvades funkcijā ievieto $\mathbf{s = j\omega}$**,** kur $\boldsymbol{\omega}$ – svārstību leņķiskā frekvence, rad/s.

Ievietojot pārvades funkcijā (5.2): $\mathbf{s = j\ \omega}$, iegūstam **vaļējas AVS frekvenču funkciju:**

$$W(j\omega) = \frac{K}{a_0 \cdot (j\omega)^2 + a_1 \cdot (j\omega) + 1} = \frac{K}{(1 - a_0 \cdot \omega^2) + j(a_1 \cdot \omega)}. \qquad (5.6)$$

Lai atbrīvotos no kompleksa skaitļa saucējā, skaitītāju un saucēju reizina ar saistītu kompleksu skaitli: **(1 - a_0·ω^2) – j(a_1·ω).** Tad saucējā iegūst reālu skaitli: **(1-a_0·ω^2)^2 + (a_1·ω)^2**.

Izdarot darbības ar kompleksiem skaitļiem, iegūstam **frekvenču funkcijas** izvērstu izteiksmi:

$$W(j\omega)=\frac{K\cdot[(1-a_0\cdot\omega^2)-j(a_1\cdot\omega)]}{(1-a_0\cdot\omega^2)^2+(a_1\cdot\omega)^2}=U(\omega)-jV(\omega)=W(\omega)\cdot e^{-j\varphi(\omega)}, \qquad (5.7)$$

kur **U(ω)** - vaļējas AVS frekvenču funkcijas **reālā komponente**:

$$U(\omega)=\frac{K\cdot(1-a_0\cdot\omega^2)}{(1-a_0\cdot\omega^2)^2+(a_1\cdot\omega)^2}; \qquad (5.8)$$

V(ω) - vaļējas AVS frekvenču funkcijas **imaginārā komponente**:

$$V(\omega)=\frac{K\cdot a_1\cdot\omega}{(1-a_0\cdot\omega^2)^2+(a_1\cdot\omega)^2}; \qquad (5.9)$$

W(ω) – vaļējas AVS frekvenču funkcijas **modulis** (izejas un ieejas signālu amplitūdu attiecība, pastiprinājuma koeficients pēc amplitūdas):

$$W(\omega)=\sqrt{U(\omega)^2+V(\omega)^2}=\frac{A_{iz}(\omega)}{A_{ie}}; \qquad (5.10)$$

φ(ω) – vaļējas AVS frekvenču funkcijas **arguments** (izejas signāla fāzes leņķis):

$$\varphi(\omega)=-arctg\,\frac{V(\omega)}{U(\omega)}=-arctg\,\frac{a_1\cdot\omega}{1-a_0\cdot\omega^2}. \qquad (5.11)$$

Frekvenču funkcija (5.7) ir kompleksa funkcija, kuru var attēlot kompleksajā plaknē ar koordinātām **[±U(ω), ±jV(ω)].** Ievietojot izteiksmē (5.7) AVS parametrus **(K, a_0, a_1)** un mainot ieejas signāla leņķisko frekvenci **ω no 0 līdz** ∞**,** iegūst **vaļējas AVS** amplitūdas – fāzes frekvenču raksturlīkni **(AFFR)** jeb **Naikvista hodogrāfu**.

Turpmāk apskatīsim AVS stabilitātes un darbības kvalitātes analīzes piemēru, pielietojot **Naikvista** un **Bodē** frekvenču raksturlīknes, kuru aprēķinam un grafiskai attēlošanai sastādīsim attiecīgu programmu Matlab komandlogā (5.2. att.).

<u>Naikvista frekvenču raksturlīknes</u>

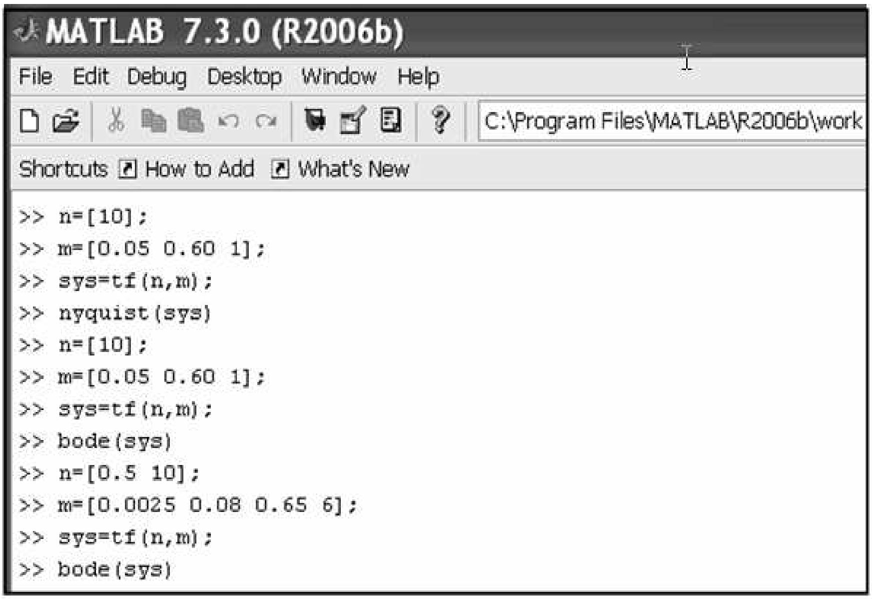
Noteiksim statiskas AVS ar inerciālu atgriezenisko saiti (5.1. att.) stabilitātes un kvalitātes rādītājus, izmantojot Naikvista hodogrāfu (5.3. att).

<u>Dots:</u> vaļējas AVS pārvades koeficients – **K = 10;** izpildiekārtas laika konstante – **T_i = 0.1 s;** vadības objekta laika konstante – **T_{obj} = 0.5 s.**

<u>Aprēķins:</u> vaļējas AVS pašoperatora koeficienti – **$a_0 = T_i \cdot T_{obj} = 0.1 \cdot 0.5 = 0.05$ s^2; $a_1 = T_i + T_{obj} = 0.1 + 0.5 = 0.6$ s.**

Lai iegūtu **Naikvista hodogrāfa** attēlu kompleksajā plaknē **[±U(ω), ±jV(ω)],** Matlab komandlogā ievadām vaļējas AVS pārvades funkcijas (5.2) **koeficientus** un Naikvista hodogrāfa aprēķina un vizualizācijas **komandfunkcijas**:

>>n=[10];→ Enter → >> m=[0.05 0.60 1];→ Enter → >> sys = tf (n, m); →Enter→ >> nyquist (sys) → Enter (5.2. att.).



5.2. att. **Matlab komandlogā sastādīta programma Naikvista un Bodē frekvenču raksturlīkņu W(jω) un L(ω), φ(ω) aprēķinam un grafiskai attēlošanai**

Matlab komandlogā iegūst vaļējas AVS amplitūdas fāzes frekvenču raksturlīkni (**AFFR)**, kas nosaukta frekvenču metodes pamatlicēja vārdā par **Naikvista hodogrāfu** (5.3. att.). To var papildināt un uzlabot, izmantojot papildus opcijas.

Dotā vaļējā AVS ir stabila, jo pašoperatora: $\mathbf{D(s) = a_0 \cdot s^2 + a_1 \cdot s + 1}$ koeficienti $\mathbf{a_0}$ un $\mathbf{a_1}$ ir pozitīvi skaitļi un tā kārta ir augstāka par iedarbes operatora $\mathbf{Q(s) = K}$ kārtu.

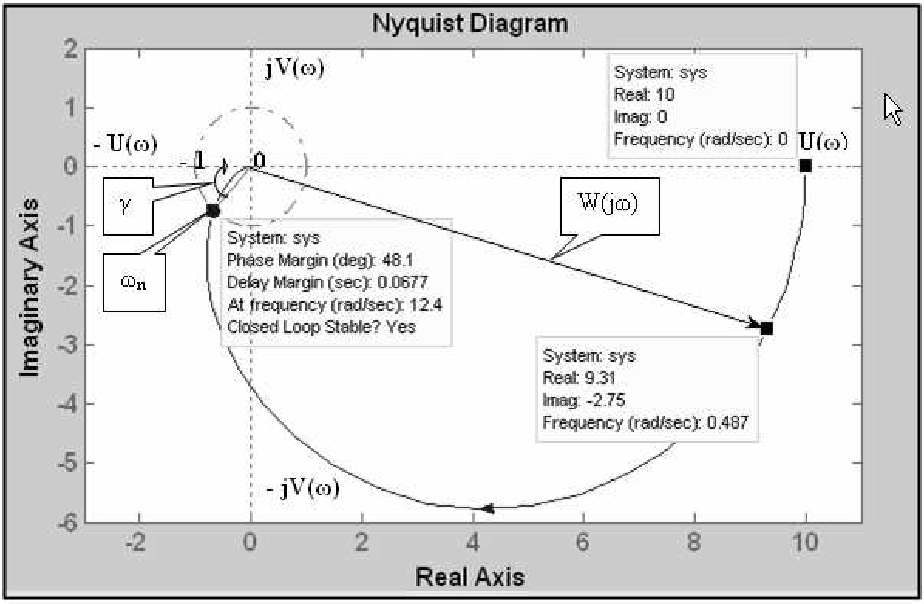
Mainoties ieejas signāla svārstību leņķiskajai frekvencei $\boldsymbol{\omega}$ no **0** līdz ∞, **Naikvista hodogrāfu** apraksta rotējoša vektora $\mathbf{W(j\ \omega) = W(\omega) \cdot e^{-\varphi(\omega)}}$ galapunkts, kurš sākas uz reālo pozitīvo skaitļu ass $\mathbf{U(\omega)}$ un pagriežas par maksimālo leņķi $\boldsymbol{\varphi(\omega = \infty) = 180^0}$. Tā kā stabilas vaļējas AVS **hodogrāfs neaptver punktu ar koordinātām (-1, j0)**, tad, atbilstoši Naikvista kritērijam, arī **slēgta AVS ir stabila**.

Vaļējas AVS pastiprinājuma koeficients jeb modulis $\mathbf{W(\omega) = A_{iz}(\omega)/A_{ie}}$ ir vienāds ar vektora $\mathbf{W(j\omega)}$ garumu: $W(\omega) = \sqrt{U(\omega)^2 + V(\omega)^2}$. un to var aprēķināt pēc vektora galapunkta koordinātām. Ja signāla leņķiskā frekvence $\boldsymbol{\omega = 0}$, tad $\mathbf{U(0)=10,}$ $\mathbf{V(0)=0}$ un $\mathbf{W(\omega)=10}$. Palielinoties $\boldsymbol{\omega}$**,** $\mathbf{W(\omega)}$ samazinās. Ja $\boldsymbol{\omega \to \infty}$**,** $\mathbf{W(\omega) \to 0}$ (5.3. att.).

Tā kā vaļējas AVS hodogrāfs nešķērso negatīvo reālo skaitļu asi: $\mathbf{-U(\omega)}$, tad stabilitātes rezerve pēc amplitūdas ir maksimāla: $\mathbf{h_{\%} = (1/1) \cdot 100\% = 100\ \%}$.

Tas nozīmē, ka dotās AVS stabilitāti un darbības kvalitāti nosaka **nogriešanas frekvence** ($\boldsymbol{\omega_n = 12.4\ s^{-1}}$), pie kuras $\mathbf{W(\omega) = 1}$, un **stabilitātes rezerve pēc fāzes**: $\boldsymbol{\gamma_{\%} = (\gamma^0/180^0) \cdot 100\% = (48.1^0/180^0) \cdot 100\% = 26.7\%}$ (sk. 5.3. att.).

Noteiksim otrās kārtas slēgtas AVS ar inerciālu atgriezenisko saiti kvalitātes rādītājus – maksimālo pārregulējumu $\boldsymbol{\sigma_{max\%}}$ un pārejas procesa laiku $\mathbf{t_p}$, izmantojot automātikas pamatos apskatītās pusempīriskās izteiksmes.



5.3. att. **Naikvista hodogrāfa W(jω) attēlojums kompleksajā plaknē stabilai statiskai otrās kārtas AVS ar pārtrauktu atgriezenisko saiti**

Maksimālais pārregulējums: $\sigma_{\max\%} \approx 93 - 2.2 \cdot \gamma_{\%} = 93 - 2.2 \cdot 26.7\% \approx 34\%.$

Pārejas procesa laiks: $t_p \approx 0.2 \cdot \dfrac{\pi \cdot (1 + 0.52 \cdot \sigma_{\max\%})}{\omega_n} = 0.2 \cdot \dfrac{3.14 \cdot (1 + 0.52 \cdot 34)}{12.4} \approx 0.95s.$

Secinājums: AVS ir stabila, taču ar nepietiekamu stabilitātes rezervi, jo maksimālais pārregulējums pārsniedz pieļaujamo – **20 %**. Lai panāktu nepieciešamo AVS darbības stabilitāti un precizitāti, jāizvēlas augstāka līmeņa vadības iekārta, piemēram, proporcionāli diferenciālais **PD – regulators**.

Bodē logaritmiskās frekvenču raksturlīknes

Iepriekš apskatītā amplitūdas fāzes frekvenču raksturlīkne **(AFFR)**: **W(jω) = W(ω)·e**$^{-j\varphi(\omega)}$ saista trīs lielumus – amplitūdu, fāzi un frekvenci. Līdz ar to amplitūdas **W(ω)** un fāzes **φ(ω)** atkarība no frekvences **ω** ir **komplekss rādītājs**, kurā katrs no atkarīgajiem mainīgajiem lielumiem iekļauts apslēptā veidā.

Lai uzskatāmi parādītu **amplitūdas** un **fāzes** atkarību no **frekvences**, **AFFR** sadala divās autonomās raksturlīknēs:

- **$W(\omega) = f_1(\omega)$** – amplitūdas frekvenču raksturlīknē **(AFR)**;
- **$\varphi(\omega) = f_2(\omega)$ -** fāzes frekvenču raksturlīknē **(FFR).**

Lai parādītu AVS izejas signāla **amplitūdas** un **fāzes** izmaiņu plašā frekvenču diapazonā, leņķisko frekvenci izsaka **logaritmiskā mērogā**. Papildus leņķiskās frekvences mērvienībai – **radiāni sekundē**, lieto arī divas citas mērvienības – **oktāvu** un **dekādi**. **Oktāvai** atbilst divkārtīga frekvences izmaiņa (**no ω līdz 2ω**), piemēram, **$\omega_1 = 10\ s^{-1}$, $\omega_2 = 20\ s^{-1}$**, bet **dekādei** – desmitkārtīga frekvences izmaiņa (**no ω līdz 10 ω**), piemēram, **$\omega_1 = 10^2\ s^{-1}$, $\omega_2 = 10^3\ s^{-1}$.**

Oktāvai un **dekādei**, kā leņķiskās frekvences mērvienībām, ir nozīmīga loma AVS sintēzes uzdevumu risināšanā, projektējot atbilstošas korekcijas iekārtas.

AVS izejas signāla fāzes nobīdes leņķi **$\varphi(\omega)$** attiecībā pret ieejas signālu izsaka **leņķa grādos**, bet izejas un ieejas signālu amplitūdu attiecību jeb **moduli $W(\omega) = A_{iz}(\omega)/A_{ie}$ – decibelos**.

Logaritmiskās amplitūdas frekvenču raksturlīknes **(LAFR)** moduļa **$W(\omega)$** mērvienība – **decibels** aizgūta no akustikas un sakaru tehnikas. Tajās divu signālu jaudu salīdzināšanai izmanto mērvienības – **Bels (B)** un **decibels (dB)** (**1 B = 10 dB**).

Automātiskās vadības praksē parasti kā informācijas nesējs tiek izmantota signālu amplitūda (strāvas stiprums, spriegums, temperatūra u.tml.), nevis jauda, tādēļ, lai saglabātu akustikā un automātiskās vadības teorijā lietoto mērvienību fizikālo ekvivalenci, logaritmisko amplitūdas frekvenču funkciju **$L(\omega)$,** kas apraksta **LAFR**, izsaka **decibelos** izmantojot pārrēķina koeficientu – **20**:

$$L(\omega) = 20 \cdot \lg \frac{A_{iz}(\omega)}{A_{ie}} = 20 \cdot \lg W(\omega). \tag{5.12}$$

Ja AVS iedarbes operators **$Q(s) = K$**, tad pie frekvences **$\omega = 0$** modulis **$W(0) = =K$**. Izsakot moduli decibelos, iegūstam $L(0) = 20 \cdot \lg K$.

<u>**Moduļa pārrēķina piemēri**</u>

1. **$W(\omega = 0) = K = 10$, $L(0) = 20 \cdot \lg K = 20 \cdot \lg 10 = 20$ dB.**
2. **$W(\omega = \omega_n) = 1$, $L(\omega_n) = 20 \cdot \lg W(\omega_n) = 20 \cdot \lg 1 = 0$ dB.**
3. **$W(\omega_1 > \omega_n) = 0.1$, $L(\omega_1) = 20 \cdot \lg W(\omega_1) = 20 \cdot \lg 0.1 = -20$ dB.**

kur ω_n – nogriešanas frekvence (s^{-1}), pie kuras **ieejas un izejas signālu amplitūdas ir vienādas**: $A_{iz}(\omega_n) = A_{ie}$.

Lai pārrēķinātu signālu amplitūdu attiecību no decibeliem **L (ω)** uz absolūtajām vērtībām **W (ω)**, izmantojam sakarību:

$$W(\omega) = 10^{\frac{L(\omega)}{20}}. \qquad (5.13)$$

Par godu AVS sintēzes frekvenču metožu pamatlicējam **H.V.Bodē** logaritmiskās amplitūdas un fāzes frekvenču raksturlīknes tiek sauktas viņa vārdā par **Bodē diagrammām** jeb **raksturlīknēm.**

Lai aprēķinātu un iegūtu Bodē raksturlīkņu grafisko attēlu vaļējai AVS, sastādām atbilstošu programmu Matlab komandlogā, ievadot vaļējas AVS pārvades funkcijas (5.2) koeficientus un Bodē raksturlīkņu aprēķina un vizualizācijas komandfunkcijas:

>>n=[10];→ Enter → >> m=[0.05 0.60 1];→ Enter → >> sys = tf (n, m); →Enter→ >> bode (sys) → Enter (5.2. att.).

Ar **„Enter”** iegūstam **Bodē** frekvenču raksturlīknes vaļējai AVS (5.4. att.), kas attēlojas **Matlab** komandlogā.

Frekvences – ω (**Frequency**), **amplitūdas – L(ω)** (**Magnitude**) un **fāzes – φ(ω)** (**Phase**) diapazoni tiek uzstādīti automātiski atbilstoši AVS inercei un jutībai. Pastāv iespēja **Bodē diagrammu** apstrādāt, pārveidot un papildināt ar informāciju, izmantojot papildus funkcijas.

Matlab programma veic ievadītās AVS pārvades funkcijas pārveidošanu Bodē logaritmiskajās frekvenču funkcijās, pēc kurām notiek frekvenču raksturlīkņu aprēķins visam automātiski uzstādītajam frekvenču izmaiņas diapazonam.

Logaritmiskā amplitūdas frekvenču raksturlīkne **(LAFR)** tiek iegūta, izmantojot atbilstošo amplitūdas frekvenču funkciju:

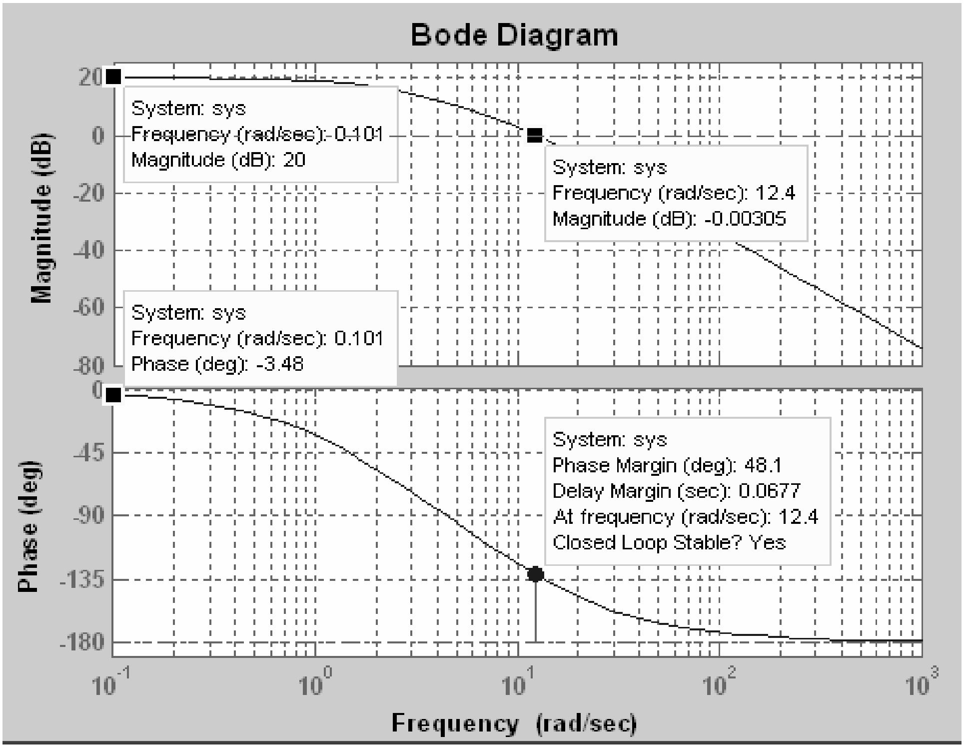
$$L(\omega) = 20\lg W(\omega) = 20\lg\frac{K}{\sqrt{(1-a_0\omega^2)^2 + (a_1\omega)^2}} = 20\lg\frac{10}{\sqrt{1+0.26\omega^2 + 0.0025\,\omega^4}},$$

kur $\mathbf{a_0 = 0.05\ s^2}$, $\mathbf{a_1 = 0.60\ s}$ – konstanti koeficienti.

Lai iegūtu fāzes frekvenču raksturlīkni **(FFR)**, izmanto atbilstošo fāzes frekvenču funkciju, kas izsaka AVS izejas signāla aizkavēšanos attiecībā pret ieejas signālu:

$$\varphi(\omega) = -arctg\ \frac{a_1 \cdot \omega}{1 - a_0 \cdot \omega^2} = -arctg\ \frac{0.60 \cdot \omega}{1 - 0.05 \cdot \omega^2}.$$

Ievietojot izteiksmēs frekvenci **ω (rad/s)**, iegūst Bodē raksturlīknes (5.4. att.).



5.4. att. **Bodē logaritmiskā amplitūdas frekvenču raksturlīkne L(ω) un fāzes frekvenču raksturlīkne φ(ω) otrās kārtas vaļējai statiskai AVS**

Visu frekvenču apgabalu var sadalīt trīs raksturīgos apgabalos:

1. zemo frekvenču diapazonā $\mathbf{(0 < \omega_z < 2\ s^{-1})}$;
2. vidējo frekvenču diapazonā $\mathbf{(3 < \omega_v < 30\ s^{-1})}$;
3. augsto frekvenču diapazonā $\mathbf{(\omega_a > 100\ s^{-1})}$.

Zemo frekvenču diapazonu nosacīti izvēlas sākot no $\omega = 0$ līdz frekvencei, pie kuras sistēmas pastiprinājuma koeficients **W(ω)** samazinās par **3 dB**.

Zemo frekvenču diapazons faktiski neiespaido AVS pārejas procesu, jo attiecas uz tā beigu posmu, kad sistēma tuvojas stacionāram režīmam. Tas galvenokārt nosaka AVS precizitāti stacionārā stāvoklī, piemēram, statisko kļūdu.

AVS **stabilitāti un darbības kvalitāti** nosaka **vidējo frekvenču diapazons ap nogriešanas frekvenci** ω_n. Dotajai sistēmai nogriešanas frekvence $\omega_n = 12.4\ s^{-1}$, kam atbilst logaritmiskais pastiprinājuma koeficients $L(\omega_n) = 0$ **dB** (5.4. att.). Vidējo frekvenču diapazonu izvēlas nosacīti apgabalā ap nogriešanas frekvenci.

Atzīmējot **nogriešanas frekvencei** atbilstošo punktu uz **fāzes frekvences raksturlīknes φ(ω)**, iegūst AVS stabilitātes rezervi pēc fāzes **(Phase Margin):**

$\gamma = 48.1^0$, kam atbilst laika rezerve **(Delay Margin):**

$$t_r = (\pi \cdot \gamma)/(\omega_n \cdot 180^0) = (3.14 \cdot 48.1^0) / (12.4 \cdot 180^0) = 0.0677\ s.$$

Stabilitātes rezervi pēc fāzes parasti izsaka procentos:

$\gamma_{\%} = (48.1^0 / 180^0) \cdot 100\% = 26.7\%$.

Rezultāti sakrīt ar tiem, kurus ieguvām izmantojot Naikvista hodogrāfu (5.3. att.).

Vaļējas AVS **LAFR** noliece pret abscisas asi pie nogriešanas frekvences ω_n dod iespēju novērtēt pārejas procesa raksturu. Šo nolieci izsaka decibelos uz oktāvu vai dekādi. Ja, piemēram, raksturlīknes noliece ir – **20 dB / dekādi**, tad pārejas process ir **monotons** vai ar **nelielu pārregulējumu**.

Dotās AVS **LAFR** noliece pie nogriešanas frekvences ir lielāka par – **20 dB / dekādi**, tāpēc pārejas process ir svārstīgs (sk. 5.4. att.).

Augstās frekvences nedaudz iespaido AVS pārejas procesa sākumu, proti, tā aizkavēšanos pirms paātrināšanās, kuru savukārt nosaka vidējās frekvences.

Analizēsim Bodē frekvenču raksturlīknes slēgtai statiskai AVS ar atgriezenisko saiti. Aptverot **otrās kārtas statisku AVS** ar **inerciālu atgriezenisko saiti**, iegūst **trešās kārtas slēgtu AVS**, kuru apraksta pārvades funkcija (5.5).

Tās iedarbes **operators**: **$Q(s) = b_0s+b_1$; pašoperators:**

$$D(s) = a_0s^3+a_1s^2+a_2s+a_3,$$

kur **b_0=0.5 s; b_1=10; a_0=0.0025 s^3; a_1=0.08 s^2; a_2 =0.65 s; a_3=6** – konstanti koeficienti.

Ievietojot pārvades funkcijā (5.5) **s = jω** un izdarot matemātiskus pārveidojumus, iegūstam frekvenču funkciju, kas apraksta **slēgtas AVS AFFR**:

$$\Phi(j\omega) = \Phi(\omega)\cdot e^{-j\varphi s(\omega)}, \qquad (5.14)$$

kur **$\Phi(\omega) = A_{izs}(\omega)/A_{ie}$** – slēgtas AVS modulis jeb pastiprinājuma koeficients;

$A_{izs}(\omega)$ – slēgtas AVS izejas signāla amplitūda;

$\varphi_s(\omega)$ - slēgtas AVS arguments jeb fāze, rad.

Bodē raksturlīknes slēgtai AVS parādītas 5.5. attēlā. Lai tās iegūtu, Matlab komandlogā ievadām slēgtas AVS pārvades funkcijas (5.5) koeficientus un Bodē raksturlīkņu aprēķina un vizualizācijas komandfunkcijas (5.2. att.):

>>n=[0.5 10];→ Enter → >> m=[0.0025 0.08 0.65 6];→ Enter

>> sys = tf (n, m); →Enter→ >> bode (sys) → Enter (5.2. att.).

Matlab komandlogā iegūstam **Bodē diagrammu**, kas sastāv no slēgtas AVS logaritmiskās amplitūdas frekvenču raksturlīknes **(LAFR) – $L_s(\omega) = 20\cdot\lg \Phi(\omega)$, dB** un fāzes frekvenču raksturlīknes **(FFR) – $\varphi_s(\omega) = f(\omega)$, leņķa grādi**,

kur leņķiskā frekvence **ω (rad/s)** atlikta logaritmiskā mērogā (5.5. att.).

Slēgtas AVS **LAF** raksturlīknei raksturīgs maksimuma pīķis: **$L_s(\omega_r)$ = 8.69 dB** pie frekvences **ω_r = 8.54 s^{-1}**, ko sauc par rezonanses frekvenci. Pie **ω = 0** logaritmiskais pastiprinājuma koeficients: **$L_s(0)$ = 4.44 dB** ir minimāls.

Izmantojot sakarību (5.13), noteiksim slēgtas AVS absolūtos **pastiprinājuma koeficientus:**

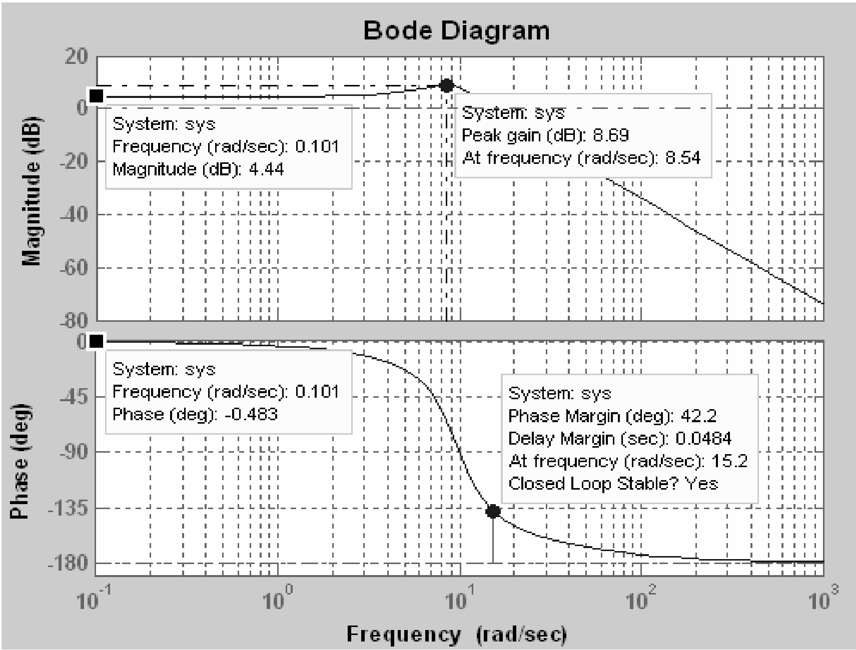
$$\Phi(0) = 10^{\frac{L_s(0)}{20}} = 10^{\frac{4.44}{20}} = 1.66,$$

$$\Phi(\omega_r) = 10^{\frac{L_s(\omega_r)}{20}} = 10^{\frac{8.69}{20}} = 2.73\,.$$

Slēgtas AVS stabilitātes rezervi un kvalitāti var novērtēt ar svārstīguma rādītāju:

$$\mathbf{M = \Phi(\omega_r)/\Phi(0) = 2.72/1.67 = 1.63.}$$

AVS stabilitātes rezerve ir pietiekama un **kvalitāte ir laba**, ja **1.1 ≤ M ≤ 1.5.** Redzam, ka dotās AVS kvalitāte nav pietiekami laba. Rezonanses pīķis norāda, ka pārejas process ir svārstīgs. Jo lielāks **M**, jo AVS ir svārstīgākā. **Ja M ≤ 1**, pārejas process ir monotons – bez pārregulējuma.



5.5. att. **Bodē logaritmiskā amplitūdas frekvenču raksturlīkne $L_s(\omega)$ un fāzes frekvenču raksturlīkne $\varphi_s(\omega)$ slēgtai AVS ar inerciālu atgriezenisko saiti**

Izmantojot slēgtas AVS logaritmisko amplitūdas frekvenču raksturlīkni $\mathbf{L_s(\omega)}$, var aptuveni novērtēt svārstīga pārejas procesa kvalitātes rādītājus.

<u>**Maksimālais pārregulējums**</u>**:**

$$\sigma_{\max}\% \approx \frac{0.8\cdot\Phi(\omega_r)-\Phi(0)}{\Phi(0)}\cdot 100\% = \frac{0.8\cdot 2.73-1.66}{1.66}\cdot 100\% = 31.6\%.$$

Pārejas procesa laiks: $t_p \approx \dfrac{(0.9-1)\pi \cdot \Phi(\omega_r)}{\omega_r} = \dfrac{0.95 \cdot 3.14 \cdot 2.73}{8.54} = 0.95s$.

Secinājumi: Aprēķini pēc Bodē raksturlīknēm sakrīt ar rezultātiem, ko ieguvām no Naikvista hodogrāfa analīzes. AVS stabilitāte un darbības kvalitāte nav apmierinoša. Jāveic pasākumi tās uzlabošanai.

5.2. Tvaika katla vadības sistēmas modelēšana

Tvaika katli, kas uzstādīti pārtikas uzņēmumos darbojas ar mainīgu neorganizētu slodzi. Tvaiks tiek izmantots **telpu apkurei**, tehnoloģiskā **karstā ūdens ieguvei** un produkcijas **sterilizācijai,** un **pasterizācijai**. Jaudīgu sterilizatoru ieslēgšana rada lielu tvaika patēriņu, kas izraisa tvaika spiediena krišanos. Pārslodzes gadījumā degļi, strādājot ar maksimālo jaudu, nespēj uzturēt tvaika spiedienu uzdotajā līmenī.

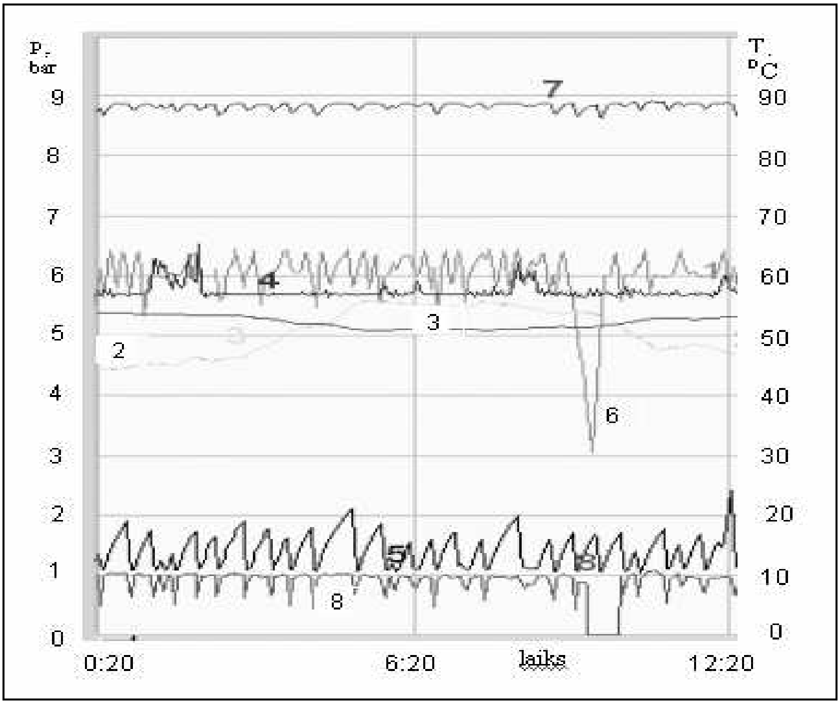
Lai izpētītu tvaika spiediena stabilizācijas problēmu un atrastu tās racionālus risinājumus, jāveic pārejas procesu modelēšana tvaika katlā. Šai nolūkā sastāda tvaika katla matemātisko modeli, izmantojot tvaika spiediena **p (bar)** un tvaika patēriņa **q (kg/min)** datorizētās uzskaites un arhivācijas datus. Galvenā perturbācija – slodze tiek modelēta ar atsevišķu kanālu **q → p** (**tvaika patēriņš – tvaika spiediens**), kas dod iespēju pētīt tvaika spiediena pārejas procesu atkarībā no slodzes lieluma un izmaiņas rakstura.

Lai imitētu tvaika katla darbību ražošanas apstākļos, sastāda tvaika spiediena regulēšanas sistēmas **algoritmisko blokshēmu**, kas dod iespēju veikt tvaika spiediena pārejas procesu imitāciju modelēšanu **Windovs** vidē, izmantojot **Matlab** apakšprogrammu **Simulink** un izpētīt iespējamos risinājumus tvaika ražošanas iekārtas darbības kvalitātes uzlabošanai.

Modelēšanas objekts un metodika

Pētījumu objekts ir **Vapor TTK-70** tvaika katls, kas uzstādīts pārtikas ražošanas uzņēmumā **"Spilva"**, un tā **automātiskās vadības sistēma** tvaika spiediena stabilizācijai mainīgas slodzes apstākļos. Tvaika katla nominālā jauda ir **1.9 MW**, ražīgums **3 t/h**.

Tvaika katla **Vapor** darbības diagramma parāda, ka vienlaicīgi darbojoties vairākiem patērētājiem tvaika spiediens svārstās robežās **6±0.5 bāri** (5.6. att.).



5.6. att. **Tvaika katla "Vapor" TTK-70 darbības parametru diagramma:**
1 – tvaika spiediens, bar; 2 – apkures kontūra ūdens temperatūra,°C; 3 – apkures kontūra ūdens aprēķinātā temperatūra, °C; 4 – karstā tehnoloģiskā ūdens temperatūra, °C; 5 – barošanas ūdens temperatūra, °C; 6 – tvaika spiediena krišanās pie liela patēriņa, bar; 7 – ūdens temperatūra deaeratorā, °C; 8 – spiediens deaeratorā, bar

Ieslēdzot jaudīgu sterilizatoru, tvaika patēriņš ievērojami pārsniedz tvaika katla nominālo ražīgumu, tādēļ spiediens nokrītas līdz **3 bāriem**. Tālāka spiediena pazemināšanās izbeidzas, jo ,uzkarstot sterilizatoram, tvaika patēriņš samazinās un regulēšanas sistēma spēj pacelt spiedienu uzdotajā līmenī. Taču lielais spiediena kritums sterilizatora uzkaršanas periodā negatīvi iespaido tvaika kvalitāti un traucē pārējo patērētāju darbību.

Tā kā veikt organizētus eksperimentus ražošanas apstākļos parasti nav iespējams, tad problēmas pētīšanai izmanto modelēšanas metodi. Šai nolūkā sastāda tvaika katla matemātisko modeli, izmantojot tā darbības datorizētās kontroles datus. Datu analīze parāda, ka tvaika spiediena regulēšanas kanāla **Q → p** (**siltuma plūsma → tvaika spiediens**) pārejas procesu pie konstantas slodzes **q = const** visadekvātāk var aprakstīt ar kavējuma posma un pirmās kārtas inerciāla aperiodiska posma virknes slēgumu.

Kavējuma posmu apraksta eksponenciāla pārvades funkcija, kas rada problēmas tvaika spiediena regulēšanas sistēmas operatorvienādojuma atrisināšanai. Tādēļ eksponentfunkciju aizstāj ar tuvinātu polinomālu izteiksmi, pielietojot **Padē aproksimāciju.** Analīze parāda, ka apmierinošus rezultātus iegūst ar otrās kārtas modificētu Padē aproksimāciju **(k = 2)**.

Tvaika katla pārejas procesu tad apraksta sekojošs **operatorvienādojums**:

$$T_k p(s)s + p(s) = K_k Q(s) \frac{1 - \frac{\tau_k}{2} s + \frac{\tau_k^2}{9} s^2}{1 + \frac{\tau_k}{2} s + \vartheta \frac{\tau_k^2}{9} s^2}, \qquad (5.15)$$

kur **p(s)** - tvaika spiediena attēlfunkcija;

Q(s)- degļu siltuma jaudas attēlfunkcija;

$\mathbf{K_k}$ - tvaika katla pārvades koeficients, **bar/MW**;

$\mathbf{T_k}$ - tvaika katla spiediena izmaiņas laika konstante, **min**;

$\boldsymbol{\tau_k}$ - tvaika katla transportkavējums, kuru rada ūdens iztvaikošanas inerce, **min**.

Programma **Matlab "Simulink"** minēto procedūru realizē automātiski, uzstādot nepieciešamo Padē approksimācijas kārtu. Parasti izvēlas **k = 4**.

Kombinētās vadības sistēmas modelis mainīgai slodzei

Modeļa algoritmiskā blokshēma (5.7. att.) sastādīta izmantojot reālās vadības sistēmas iekārtu parametrus. Tajā ietilpst firmas **Honeywell** brīvi programmējamais adaptīvais **mikrokontrolleris UMC 800** un **Oilon** tipa regulējama universāla energģētiskā iekārta **GKP-400M** ar siltuma jaudu no **1.15** līdz **3.6 MW** , kas paredzēta gāzveida un šķidram kurināmajam.

Mikrokontrolleris nodrošina tvaika spiediena automātisku regulēšanu pēc **proporcionāli integrālā – diferenciālā (PID)** algoritma ar maksimālā pārregulējuma efektīvu ierobežošanu, aktivizējot **fuzzy** loģikas bloku.

Ideālo **PID** algoritmu tehniski realizēt nav iespējams. Mikrokontrolleros izmanto modificēto **PID** vadības algoritmu, kuru visadekvātāk apraksta sekojoša **pārvades funkcija:**

$$W_{PID} = K_p \left(1 + \frac{1}{T_i .s}\right)\left(\frac{T_d .s + 1}{T_f .s + 1}\right), \qquad (5.16)$$

kur K_p - proporcionālās ķēdes pārvades koeficients;

T_i - integrējošās ķēdes laika konstante, **min**;

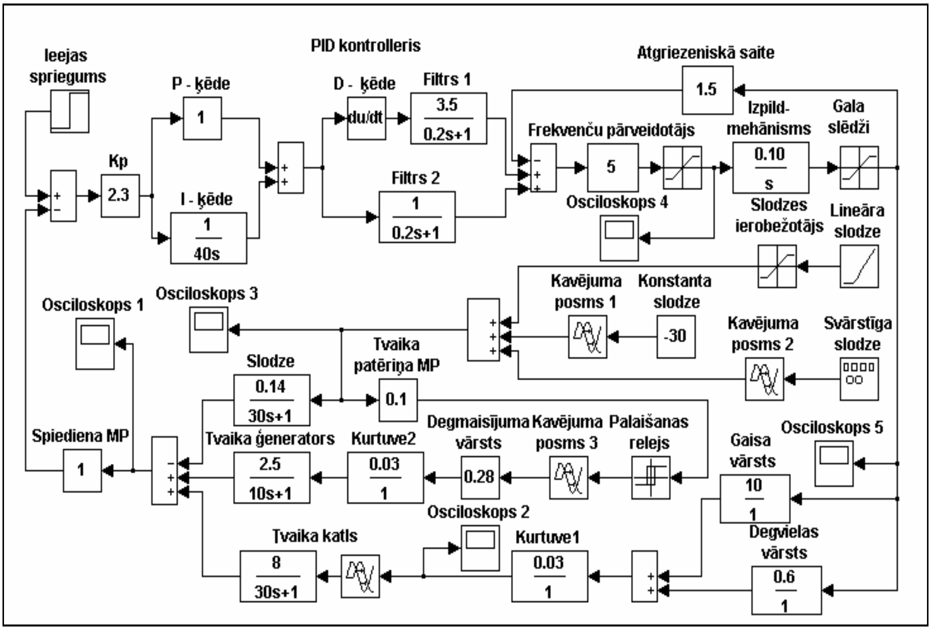
T_d - diferencējošās ķēdes laika konstante, **min**;

T_f **= (0,05 – 0,1)** T_d - filtra laika konstante, **min**.

Atbilstoši pārvades funkcijai (5.16) sastādīta **PID kontrollera** modelēšanas blokshēma (5.16. att.). Tā parametru $\mathbf{K_p}$**,** $\mathbf{T_i}$ un $\mathbf{T_d}$ iestatījumus nosaka pēc **Cīglera – Nikolsa** kritērijiem. Taču ņemot vērā, ka reāla sistēma satur nelineārus elementus (relejus, galaslēdžus, ierīces ar piesātinājumu un transportkavējumu), aprēķinātie iestatījumi var būtiski atšķirties no optimālajiem. Tādēļ nepieciešams veikt parametru $\mathbf{K_p}$, $\mathbf{T_i}$ un $\mathbf{T_d}$ precizējumus modelēšanas gaitā atbilstoši vadības objekta pārejas procesa kvalitātei.

Tvaika katla vadības sistēmas algoritmiskā blokshēma sastāv no tvaika spiediena stabilizācijas kontūra, kuru veido **PID kontrolleris**, **Frekvenču pārveidotājs**, integrējošs **Izpildmehānisms** ar stabilizējošu **Atgriezenisko saiti** un **Galaslēdžiem Degvielas** un **Gaisa** regulēšanas **vārstu** kopējās vārpstas pagrieziena leņķa ierobežošanai, **Kurtuves**, **Tvaika katla** un **Spiediena mērīšanas pārveidotāja**, kas padod spiedienam proporcionālu spriegumu uz sistēmas ieeju (5.7. att.).

Katla palaišanas komandu formē ar kāpņveida **Ieejas sprieguma** ģeneratoru , ar kuru iestata uzdoto tvaika spiedienu. Spiediena regulēšanas raksturlīkni **p = f(t)** iegūst uz **Osciloskopa 1** ekrāna. Pārējie **Osciloskopi 2, 3, 4** parāda attiecīgo sistēmas posmu reakciju uz ieejas iedarbēm.



5.7. att. **Tvaika katla kombinētās vadības sistēmas algoritmiskā blokshēma**

Tvaika katla reakciju uz tvaika patēriņa izmaiņu modelē bloks **Slodze**. Tvaika patēriņa slodzes statiskās un dinamiskās komponentes formē ar blokiem **Lineāra slodze**, kuras maksimālo lielumu iestata ar **Slodzes ierobežotāju, Konstanta slodze**, kuru aktivizē ar bloku **Kavējuma posms 1**, un **Svārstīga slodze**, kuru aktivizē ar bloku **Kavējuma posms 2**.

Lai pētītu pārslodzes uz tvaika katlu iespaida ierobežošanas iespējas, shēmā iekļauts kontūrs ar ātrdarbīgu rezerves **Tvaika ģeneratoru**, kas automātiski aktivizējas, ja tvaika patēriņa slodze ilgstoši pārsniedz **60 kg/min**. Tvaika ģeneratora palaišanas kavējumu iestata ar **Kavējuma posmu 3**.

Ja **Tvaika patēriņa mērīšanas pārveidotājs** ilgstoši signalizē par pārslodzi, tad ieslēdzas **Palaišanas relejs**, kas iedarbina degmaisījuma degļus un palaiž tvaika ģeneratoru. Tvaika ražošanas procesa vadība notiek pēc diviem parametriem – tvaika spiediena **p** un tvaika patēriņa **q**, kas būtiski uzlabo tvaika iekārtas darbību svārstīgas slodzes un pārslodžu gadījumos.

Modelēšanas rezultātu analīze

Atbilstoši modelēšanas algoritmiskajai struktūrshēmai pārejas procesus tvaika katla kombinētās vadības sistēmā apraksta sekojošs **operatorvienādojums**:

$$p(s) = \frac{W_g(s) \cdot U_{ie}(s) + [W_{tg}(s) - W_q(s)] \cdot q(s)}{1 + W_g(s) \cdot K_s}, \qquad (5.17)$$

kur $\mathbf{U_{ie}(s)}$ – sistēmas ieejas sprieguma attēls;

q (s) – tvaika patēriņa (slodzes) attēls;

$\mathbf{W_g(s)}$ – tvaika katla vadības kontūra bez atgriezeniskās saites pārvades funkcija;

$W_{tg}(s)$ – tvaika ģeneratora vadības ķēdes pārvades funkcija;

$\mathbf{W_q(s)}$ – slodzes iedarbes pārvades funkcija;

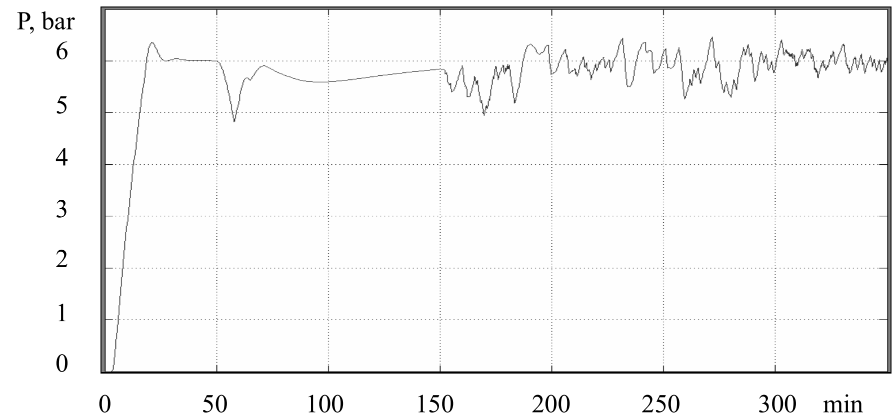
$\mathbf{K_s}$ - spiediena mērīšanas pārveidotāja pārvades koeficients, **V/bar**.

Operatorvienādojums (5.17) apraksta tvaika spiediena izmaiņas procesu **p = f(t)**, kura attēlu simulācijas beigās iegūst uz **Osciloskopa 1** ekrāna (5.8. att.).

Tvaika patēriņa **q(t)** raksturlīkni iegūst uz **Osciloskopa 3** ekrāna (5.9. att.).

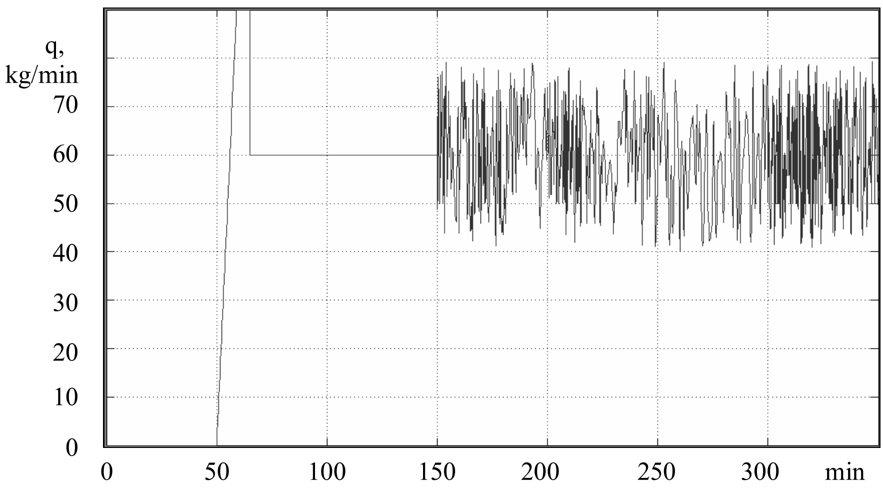
Tvaika katla palaišana tiek modelēta tukšgaitas režīmam. Pēc 50 minūtēm, ko iestata **Lineārās slodzes** formēšanas blokā, tiek iedarbināts sterilizators, un tvaika patēriņš sāk pieaugt ar paātrinājumu **10 kg/min**^{2}. Kad patēriņš sasniedz **60 kg/min,** ieslēdzas rezerves tvaika ģeneratora **Palaišanas relejs**. **Kavējuma posms3** darbojas kā laika relejs, kas novērš tvaika ģeneratora iedarbināšanu no īslaicīgām pārslodzēm.

Ar slodzes ierobežotāju tiek iestatīts slodzes maksimālais lielums **90 kg/min**, kas par **80%** pārsniedz dotā katla nominālo ražīgumu **50 kg/min**. Sakarā ar tvaika ģeneratora inerci spiediens nokrītas zem **5 bāriem** (5.8. att.).



5.8. att. **Tvaika katla palaišanas un spiediena regulēšanas raksturlīkne p = f(t)**

Iedarbinoties tvaika ģeneratoram spiediens sāk paaugstināties. Uzkarstot sterilizatoram, tvaika patēriņš jūtami samazinās. Šo atslodzi formē ar bloku **Konstanta slodze** (5.9. att.). Sistēmā rodas neliels pārregulējums, kura beigās iestatās uzdotais tvaika spiediens **p_0 = 6 bāri.**



5.9. att. **Tvaika patēriņa slodzes modelēšanas raksturlīkne q = f(t)**

Lai modelētu svārstīgas slodzes iespaidu uz tvaika katla darbību, konstantajai slodzei **60 kg/min** uzklāj gadījuma rakstura mainīgu komponenti ar izmaiņas amplitūdu **±20 kg/min** un frekvenci **0.05 min^{-1}**. Šai nolūkā **150-tajā** minūtē tiek aktivizēts gadījuma skaitļu ģenerators **Svārstīga slodze**.

Raksturlīkņu (5.8., 5.9. att.) salīdzinājums parāda, ka apskatītā tvaika katla kombinētās vadības sistēma nodrošina tvaika spiediena pietiekamu stabilizāciju neorganizētas svārstīgas slodzes un ilgstošu pārslodžu gadījumos.

LITERATŪRA

1. A. Šnīders. Automātiskās vadības pamati.– Jelgava: LLU, 2007. – 220 lpp.
2. A. Šnīders. Kokapstrādes automatizācija. – Rīga: Avots, 1989. – 158 lpp.
3. V. Klimavičius. Automātiskā vadība. – Rīga: RTU, 2002. – 231 lpp.
4. I. Raņķis. Regulēšanas teorijas pamati. – Rīga: RTU, 1998. – 90 lpp.
5. J.Osis. Automātiskā vadība un regulēšana.– R.: Zvaigzne, 1969. – 267 lpp.
6. A.Šulcs. Datortehnika un ražošanas automatizācija. – Rīga: RTU, 1995. – 103 lpp.
7. E. Dzelzītis. Siltuma, gāzes un ūdens inženiersistēmu automatizācijas pamati. - Rīga: Gandrs, 2005. – 414 lpp.
8. J.Greivulis, I. Raņķis. Iekārtu vadības elektroniskie elementi un mezgli. – Rīga: Avots, 1997. – 288 lpp.
9. Finn Haugen. Modelling and control of dynamic systems.- Skien, Norway: Control Consult, 1997. – 242 p.
10. C.Smith, A.Corripio. Principles and Practice of Automatic Process Control. – New York: John Willey & Sons Inc., 1997. – 768 p.
11. Луье Б.Я., Энрайт П.Дж. Классические методы автоматического управления / Под ред. А.А.Ланнэ. – СПб.: БХВ – Петербург, 2004. – 640 с.
12. Наладка средств автоматизации и автоматических систем регулирования – Справочное пособие / Под ред. А.С.Клюева. – М.: Энергоатомиздат, 1989 – 368 с.
13. Половко А.М., Бутусов П.Н. MATLAB для студента. – СПб.: БХВ – Петербург, 2005. – 320 с.
14. Черных И.В. SIMULINK: среда создания инженерных приложений – М.: Диалог – МИФИ, 2004. – 496 с.
15. Гультяев А. К. Имитационное моделирование в среде Windows. Практическое пособие. – Санкт-Петербург: Корона Принт, 2001. – 400 с.