



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Mechatronika, Optika és Gépészeti Informatika Tanszék

Digitális szervohajtások - (BMEGEMIMMDS)
Visszacsatolt kör mérések - Motion control/Exercise

5

Név: Szőke Kálmán Benjamin
Neptun: SLZ0UE
Dátum: 2015. május 10.

Tartalomjegyzék

1. Mérési feladatok	1
1.1. Módszerek a PID szabályozó paramétereinek megválasztásához, Ziegler-Nichols hangolás	1
1.2. P szabályozó	3
1.3. P szabályozó változó alapjellel	4
1.4. P szabályozó terhelésrádobással	6
1.5. PI szabályozó	8
1.6. PI szabályozó változó alapjellel	10
1.7. PI szabályozó terhelésrádobással	12
1.8. Anti windup PI szabályozó	14
1.9. Időkésleltetett rendszer PI szabályozása	16
1.10. P és PI szabályozó ugrás függvény pozíció referenciajel válasza	19
1.11. Stick-slip jelenség	22
1.12. Pozíció szabályozás belső sebesség szabályozóval	25
1.13. Csúszómód szabályozó	29
2. Melléklet	38

Ábrák jegyzéke

1. Ziegler-Nichols hangolás	1
2. P szabályozó működése, $\Omega = 7$ rad/s	3
3. P szabályozó működése változó alapjellel	4
4. P szabályozó terhelésrádobással	6
5. PI szabályozó, $\Omega = 8$ rad/s	8
6. PI szabályozó változó alapjellel	10
7. PI szabályozó terhelésrádobással	12
8. PI szabályozó hibatoleranciája	14
9. PI szabályozó hibatoleranciája, Integrálás	14
10. Időkésleltetett PI szabályozó, Sebesség	16
11. Időkésleltetett PI szabályozó, Nyomaték	17
12. Időkésleltetett PI szabályozó, Integrálás	17
13. P szabályozó a referencia pozíció ugrásszerű változására, $P = 1, I = 0$. . .	19
14. P szabályozó a referencia pozíció ugrásszerű változására, $P = 0.5, I = 0$. .	19
15. P szabályozó a referencia pozíció ugrásszerű változására, $P = 0.5, I = 0.3$.	20
16. Stick-slip jelenség, $P = 0.1, I = 0.1$	22
17. Stick-slip jelenség, $P = 0.15, I = 0.2$	23
18. Kaszkád szabályozás, Pozíció $P_{pos} = 3, P_{vel} = 2.3, I_{vel} = 50$	25
19. Kaszkád szabályozás, $P_{pos} = 3, P_{vel} = 2.3, I_{vel} = 50$	25
20. Kaszkád szabályozás, Pozíció $P_{pos} = 1.5, P_{vel} = 2.3, I_{vel} = 50$	26
21. Kaszkád szabályozás, $P_{pos} = 1.5, P_{vel} = 2.3, I_{vel} = 50$	26
22. A pozíció szabályozó eredményei csúszómódban	29
23. A pozíció szabályozó eredményei csúszómódban, Hiba	30

24.	Csúszómód szabályozó sebesség szűrővel (kotyogás)	32
25.	Csúszómód szabályozó sebesség szűrővel (kotyogás), Hiba	33
26.	Csúszómód szabályozó sebesség szűrővel (nincs kotyogás)	34
27.	Csúszómód szabályozó sebesség szűrővel (nincs kotyogás), Hiba	35

1. Mérési feladatok

1.1. Módszerek a PID szabályozó paramétereinek megválasztásához, Ziegler- Nichols hangolás

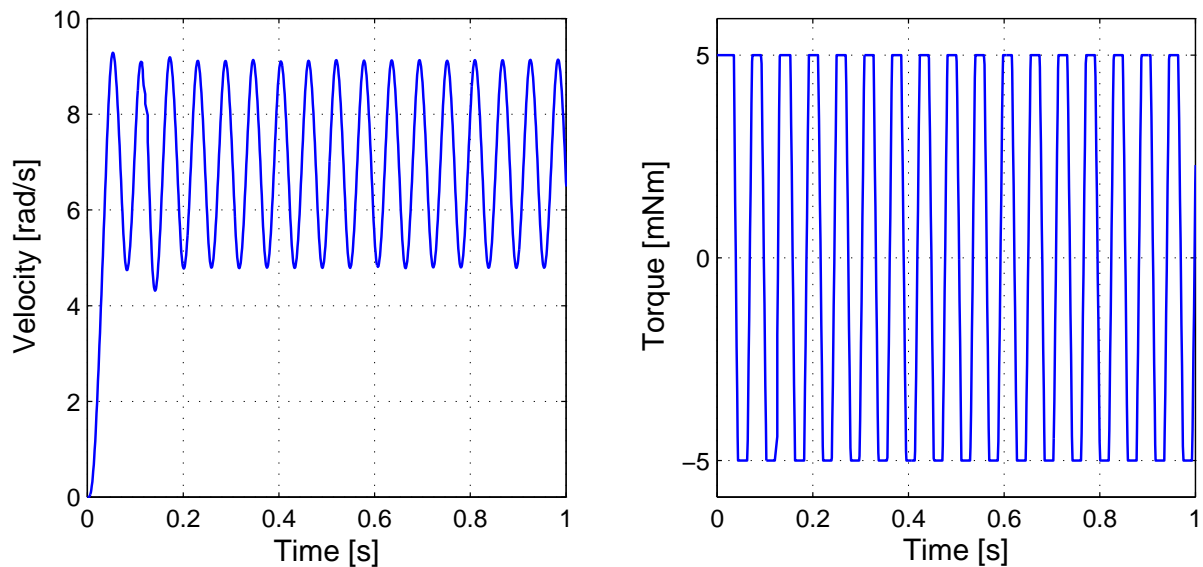
A Ziegler-Nichols hangolás során mért Au , Tu paraméterek értéke:

$$Au = 4.5$$

$$Tu = 0.058$$

	A_p	I	T_D
P	2.25	0	0
PI	2.045	46.55	0
PID	2.647	77.58	0.0163125

1. táblázat. Ziegler-Nichols hangoló táblázat



1. ábra. Ziegler-Nichols hangolás

```
// A mérés dátuma és időpontja: 2015. 05. 07. 20:41 CET
// IP Address (IPv4): 86.59.221.241
//Measurement length in milliseconds: 2000
//The state variable names:
//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
```

```

//5. ref
//6. error

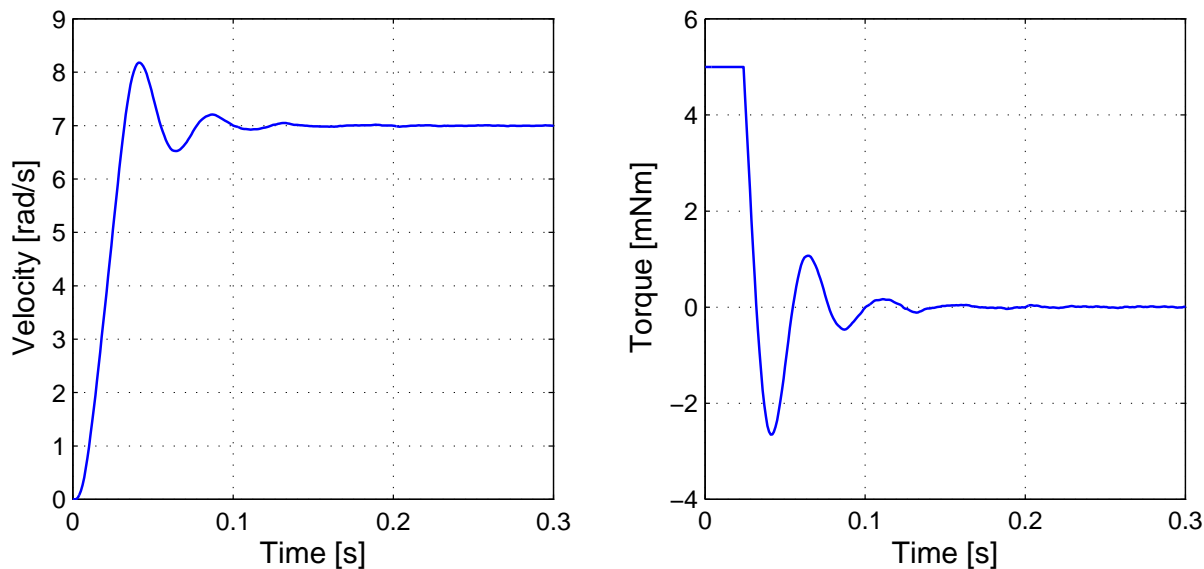
//declarations:
double P = 4.5;
double ref_vel = 7;
double error_vel= 0;
/* sebesség szűrés változók */
static float z_1=0.0, z_2=0.0, z_3=0.0;
static float ztmp_1=0.0, ztmp_2=0.0;
/* sebesség szűrés paraméterei */
/* Tsample=1e-3 and Tc=0.005 */
float ad11= 0.9989, ad12= 9.8248e-004, ad13= 4.0937e-007;
float ad21= -3.2749, ad22= 0.9497, ad23= 7.3686e-004;
float ad31= -5.8949e+003, ad32= -91.6978, ad33= 0.5076;
float bd1= 0.0011, bd2= 3.2749, bd3= 5.8949e+003;

//controller:
/* sebesség szűrés */
ztmp_1=ad11* z_1+ad12* z_2+ad13* z_3 + bd1* ResultData.Velocity;
ztmp_2=ad21* z_1+ad22* z_2+ad23* z_3 + bd2* ResultData.Velocity;
z_3=ad31* z_1+ad32* z_2+ad33* z_3 + bd3* ResultData.Velocity;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
//hibaszámítás a sebesség szabályozáshoz
error_vel=ref_vel- ResultData.Velocity;
ResultData.StateVariable_5 = ref_vel;
ResultData.StateVariable_6 = error_vel;
ResultData.Torque = P*error_vel;
if (ResultData.Torque > 5)
{
ResultData.Torque = 5;
}
if (ResultData.Torque < -5)
{
ResultData.Torque = -5;
}

```

1.2. P szabályozó

Vizsgálja meg a P szabályozó működését $\Omega = 7$ rad/s tengelysebesség mellett.



2. ábra. P szabályozó működése, $\Omega = 7$ rad/s

```
// A mérés dátuma és időpontja: 2015. 05. 07. 21:46 CET
// IP Address (IPv4): 86.59.221.241
//Measurement length in milliseconds: 1000
//The state variable names:
//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. ref
//6. error
//7. integral

//declarations:
double P_par = 2.25;
double I_par = 0.0;
double ref_vel = 7;
double error_vel;
static double error_vel_int=0.0;
double load = 2;
/* sebesség szűrés változók */
static float z_1=0.0, z_2=0.0, z_3=0.0;
static float ztmp_1=0.0, ztmp_2=0.0;
/* Tsample=1e-3 és Tc=0.0027 */
float ad11= 0.9936, ad12= 9.4621e-004, ad13= 3.4524e-007;
float ad21= - 17.5400, ad22= 0.8515, ad23= 5.6261e-004;
float ad31= -2.8584e+004, ad32= -249.0676, ad33= 0.2264;
float bd1= 0.0064, bd2= 17.5400, bd3= 2.8584e+004;
```

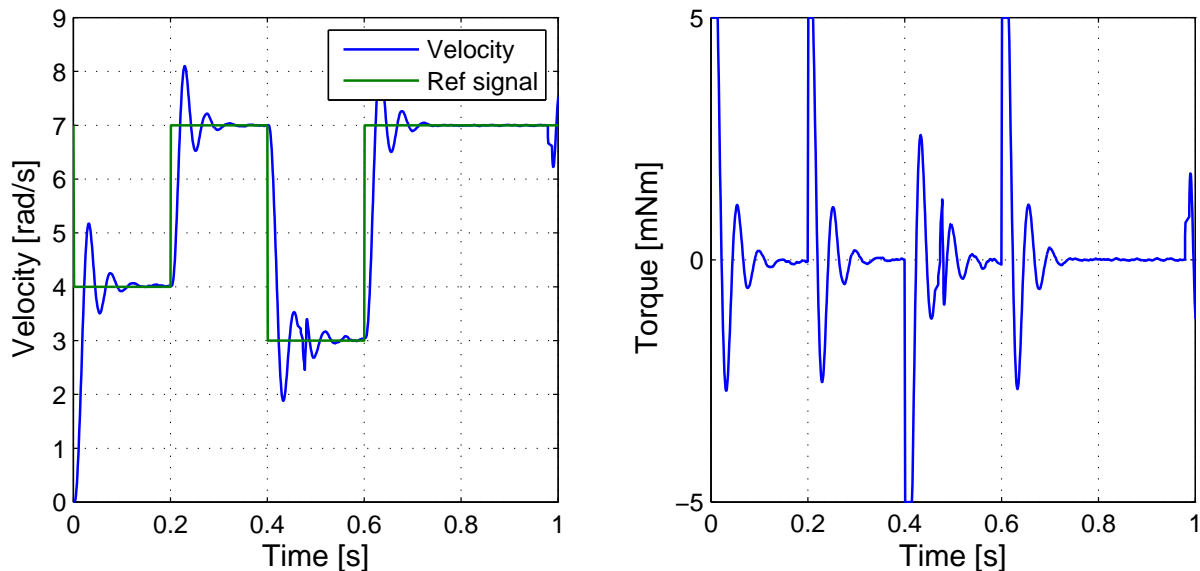
```

//controller:
if (CurrentTime >= 0.4*1e3 && CurrentTime < 0.7*1e3)
{
load = 0.5;
}
/* Sebesség szűrés */
ztmp_1=ad11* z_1+ad12* z_2+ad13* z_3 + bd1* ResultData.Velocity;
ztmp_2=ad21* z_1+ad22* z_2+ad23* z_3 + bd2* ResultData.Velocity;
z_3=ad31* z_1+ad32* z_2+ad33* z_3 + bd3* ResultData.Velocity;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
//Sebesség szabályozás hibaszámítása
error_vel=ref_vel- ResultData.Velocity;
error_vel_int = error_vel_int + error_vel*(CurrentTime -
OldTime)/1000;
ResultData.StateVariable_5 = ref_vel;
ResultData.StateVariable_6 = error_vel;
ResultData.StateVariable_7 = error_vel_int;
ResultData.Torque = P_par*error_vel + I_par*error_vel_int - load;
if (ResultData.Torque > 5) { ResultData.Torque = 5; }
if (ResultData.Torque < -5) { ResultData.Torque = -5; }

```

1.3. P szabályozó változó alapjellel

Az előző feladat kibővítése, azzal a különbséggel, hogy először $\Omega = 4$ rad/s $t = 0$ s-nál, majd $\Omega = 7$ rad/s $t = 0.2$ s-nál, $\Omega = 3$ rad/s $t = 0.4$ s-nál, és végül $\Omega = 7$ rad/s $t = 0.6$ s-nál.



3. ábra. P szabályozó működése változó alapjellel

```

// A mérés dátuma és időpontja: 2015. 05. 10. 19:01 CET
// IP Address (IPv4): 195.228.223.137
//Measurement length in milliseconds: 1000
//The state variable names:
//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. ref
//6. error
//7. integral

//declarations:
double P_par = 2.3;
double I_par = 0.0;
double ref_vel = 7;
double error_vel;
static double error_vel_int=0.0;
double load = 0;
/* sebesség szűrés változói */
static float z_1=0.0, z_2=0.0, z_3=0.0;
static float ztmp_1=0.0, ztmp_2=0.0;
/* Tsample=1e-3 és Tc=0.0027 */
float ad11= 0.9936, ad12= 9.4621e-004, ad13= 3.4524e-007;
float ad21= - 17.5400, ad22= 0.8515, ad23= 5.6261e-004;
float ad31= -2.8584e+004, ad32= -249.0676, ad33= 0.2264;
float bd1= 0.0064, bd2= 17.5400, bd3= 2.8584e+004;

//controller:
//Ugrások a referencia sebességben
if (CurrentTime > 0*1e3 && CurrentTime < 0.2*1e3)
{
ref_vel = 4;
}
if (CurrentTime >= 0.2*1e3 && CurrentTime < 0.4*1e3)
{
ref_vel = 7;
}
if (CurrentTime >= 0.4*1e3 && CurrentTime < 0.6*1e3)
{
ref_vel = 3;
}
/* Velocity filter */
ztmp_1=ad11* z_1+ad12* z_2+ad13* z_3 + bd1* ResultData.Velocity;
ztmp_2=ad21* z_1+ad22* z_2+ad23* z_3 + bd2* ResultData.Velocity;
z_3=ad31* z_1+ad32* z_2+ad33* z_3 + bd3* ResultData.Velocity;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
//sebesség szabályozás hibaszámítása
error_vel=ref_vel- ResultData.Velocity;
error_vel_int = error_vel_int + error_vel*(CurrentTime -
OldTime)/1000;
ResultData.StateVariable_5 = ref_vel;
ResultData.StateVariable_6 = error_vel;

```



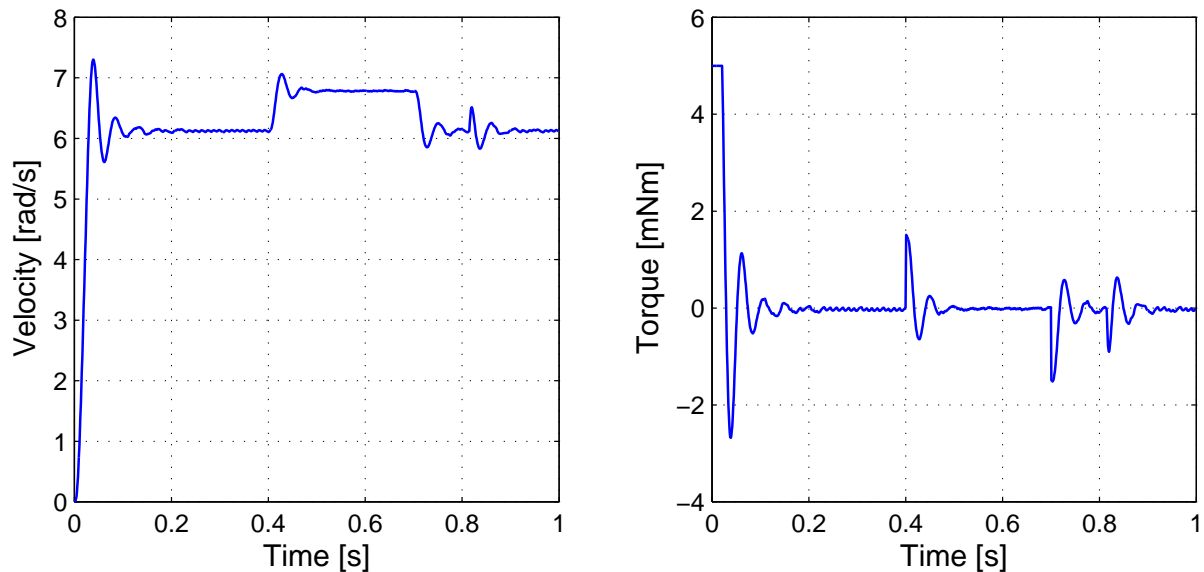
```

ResultData.StateVariable_7 = error_vel_int;
ResultData.Torque = P_par*error_vel + I_par*error_vel_int - load;
if (ResultData.Torque > 5) { ResultData.Torque = 5; }
if (ResultData.Torque < -5) { ResultData.Torque = -5; }

```

1.4. P szabályozó terhelésrádobással

Vizsgálja meg a P szabályozó zavarérzékenységét. A zavart $t = 0.5$ s-ban adjuk a rendszerre, mint negatív nyomatékot $T_{load} = -0.5$, amely kivonódik a szabályozó által számolt kimeneti feszültségből.



4. ábra. P szabályozó terhelésrádobással

```

// A mérés dátuma és időpontja: 2015. 05. 07. 20:41 CET
// IP Address (IPv4): 86.59.221.241
//Measurement length in milliseconds: 2000
//The state variable names:
//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. ref
//6. error

//declarations:
double P = 4.5;
double ref_vel = 7;
double error_vel= 0;
/* sebesség szűrés változók */
static float z_1=0.0, z_2=0.0, z_3=0.0;
static float ztmp_1=0.0, ztmp_2=0.0;

```

```

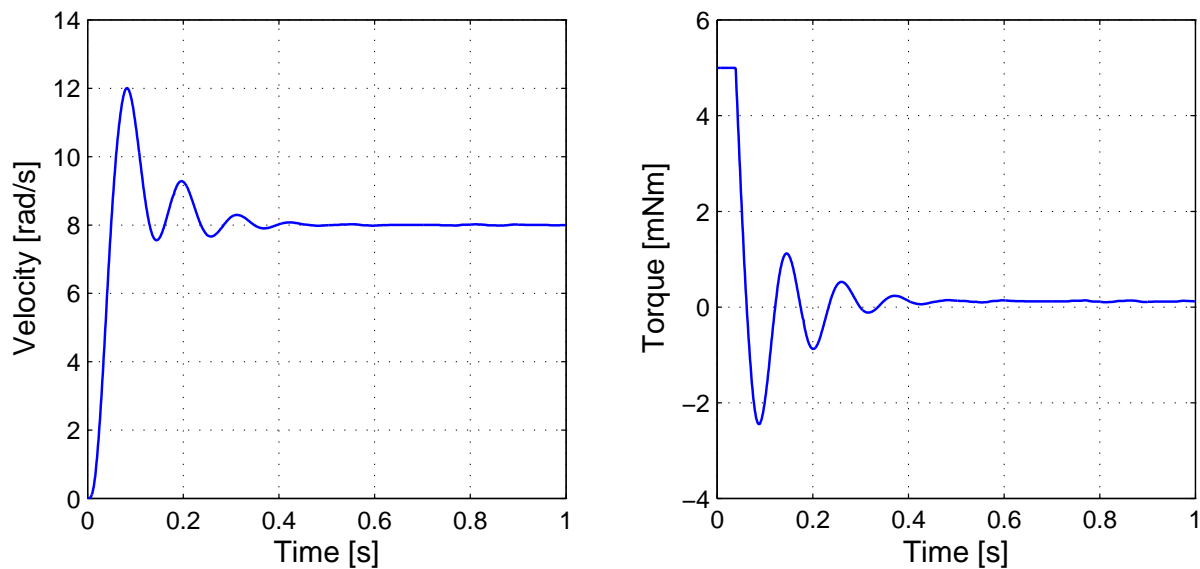
/* sebesség szűrés paraméterei */
/* Tsample=1e-3 and Tc=0.005 */
float ad11= 0.9989, ad12= 9.8248e-004, ad13= 4.0937e-007;
float ad21= -3.2749, ad22= 0.9497, ad23= 7.3686e-004;
float ad31= -5.8949e+003, ad32= -91.6978, ad33= 0.5076;
float bd1= 0.0011, bd2= 3.2749, bd3= 5.8949e+003;

//controller:
/* sebesség szűrés */
ztmp_1=ad11* z_1+ad12* z_2+ad13* z_3 + bd1* ResultData.Velocity;
ztmp_2=ad21* z_1+ad22* z_2+ad23* z_3 + bd2* ResultData.Velocity;
z_3=ad31* z_1+ad32* z_2+ad33* z_3 + bd3* ResultData.Velocity;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
//hibaszámítás a sebesség szabályozáshoz
error_vel=ref_vel- ResultData.Velocity;
ResultData.StateVariable_5 = ref_vel;
ResultData.StateVariable_6 = error_vel;
ResultData.Torque = P*error_vel;
if (ResultData.Torque > 5)
{
ResultData.Torque = 5;
}
if (ResultData.Torque < -5)
{
ResultData.Torque = -5;
}

```

1.5. PI szabályozó

Vizsgálja meg a PI szabályozó működését $\Omega = 8$ rad/s tengelysebesség mellett.



5. ábra. PI szabályozó, $\Omega = 8$ rad/s

```
// A mérés dátuma és időpontja: 2015. 05. 07. 21:48 CET
// IP Address (IPv4): 86.59.221.241
//Measurement length in milliseconds: 1000
//The state variable names:
//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. ref
//6. error
//7. integral

//declarations:
double P = 2.045;
double I = 46.55;
double ref_vel = 8;
double error_vel;
static double error_vel_int=0.0;
double load = 0;
/* sebesség szűrés változói */
static float z_1=0.0, z_2=0.0, z_3=0.0;
static float ztmp_1=0.0, ztmp_2=0.0;
/* sebesség szűrés paraméterei */
/* Tsample=1e-3 and Tc=0.007 */
float ad11= 0.9996, ad12= 9.9072e-004, ad13= 4.3344e-007;
float ad21= -1.2637, ad22= 0.9730, ad23= 8.0496e-004;
float ad31= -2.3468e+003, ad32= -50.5468, ad33= 0.6280;
```

```

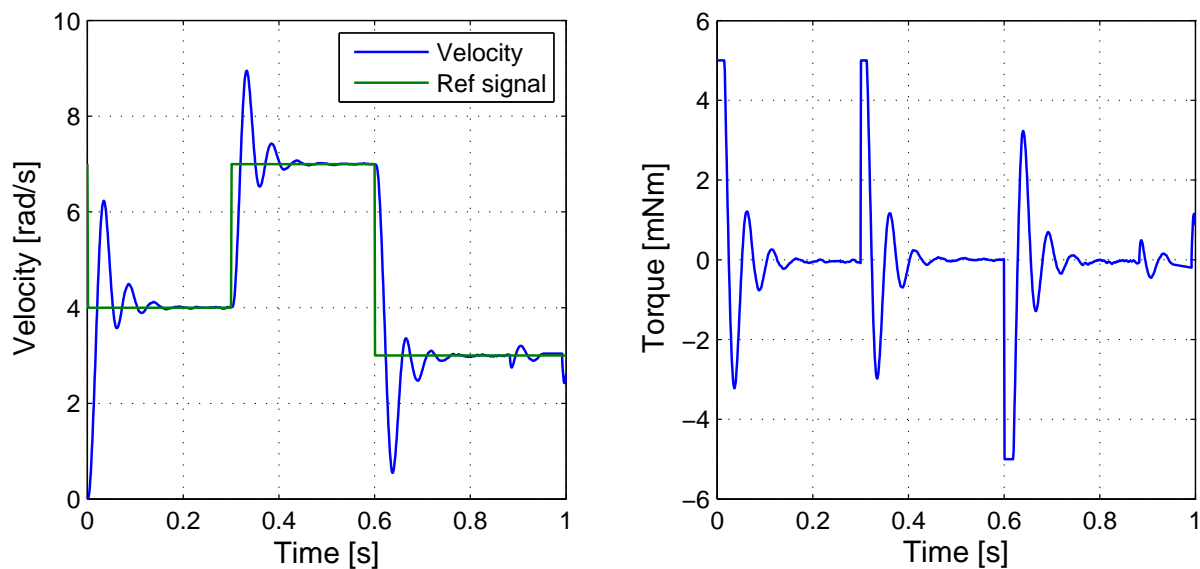
float bd1= 4.3671e-004, bd2= 1.2637, bd3= 2.3468e+003;

//controller:
/* Sebesség szűrés */
ztmp_1=ad11* z_1+ad12* z_2+ad13* z_3 + bd1* ResultData.Velocity;
ztmp_2=ad21* z_1+ad22* z_2+ad23* z_3 + bd2* ResultData.Velocity;
z_3=ad31* z_1+ad32* z_2+ad33* z_3 + bd3* ResultData.Velocity;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
//Sebesség szabályozó hibaszámítása
error_vel=ref_vel- ResultData.Velocity;
error_vel_int = error_vel_int + error_vel*(CurrentTime -
OldTime)/1000.0;
ResultData.StateVariable_5 = ref_vel;
ResultData.StateVariable_6 = error_vel;
ResultData.StateVariable_7 = error_vel_int;
ResultData.StateVariable_8 = CurrentTime;
ResultData.Torque = P*error_vel + I*error_vel_int - load;
if (ResultData.Torque > 5)
{
ResultData.Torque = 5;
}
if (ResultData.Torque < -5)
{
ResultData.Torque = -5;
}

```

1.6. PI szabályozó változó alapjellel

Legyen először $\Omega = 4$ rad/s $t = 0$ s-nál, majd $\Omega = 7$ rad/s $t = 0.3$ s-nál, és végül $\Omega = 3$ rad/s $t = 0.6$ s-nál.



6. ábra. PI szabályozó változó alapjellel

```
// A mérés dátuma és időpontja: 2015. 05. 07. 22:03 CET
// IP Address (IPv4): 86.59.221.241
//Measurement length in milliseconds: 1200
//The state variable names:
//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. ref
//6. error
//7. integral

//declarations:
double P_par = 2;
double I_par = 50.0;
double ref_vel = 7;
double error_vel;
static double error_vel_int=0.0;
double load = 0;
/* sebesség szűrés változók */
static float z-1=0.0, z-2=0.0, z-3=0.0;
static float ztmp-1=0.0, ztmp-2=0.0;
/* Tsample=1e-3 és Tc=0.0027 */
float ad11= 0.9936, ad12= 9.4621e-004, ad13= 3.4524e-007;
float ad21= - 17.5400, ad22= 0.8515, ad23= 5.6261e-004;
float ad31= -2.8584e+004, ad32= -249.0676, ad33= 0.2264;
```

```

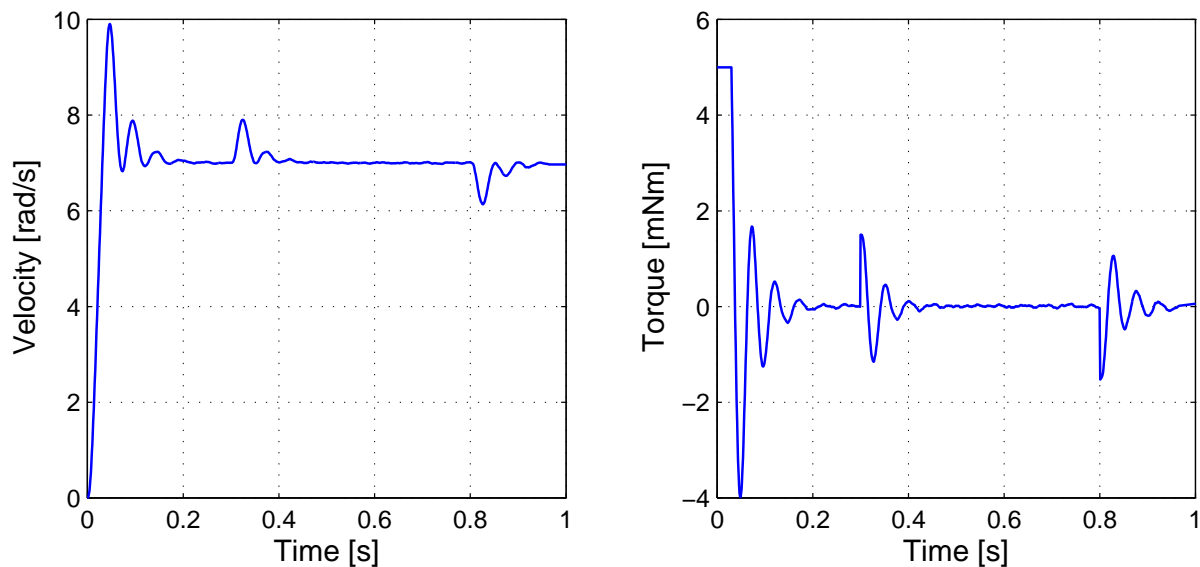
float bd1= 0.0064, bd2= 17.5400, bd3= 2.8584e+004;

//controller:
//Ugrás változások referencia sebességben
if (CurrentTime > 0*1e3 && CurrentTime < 0.3*1e3)
{
ref_vel = 4;
}
if (CurrentTime >= 0.3*1e3 && CurrentTime < 0.6*1e3)
{
ref_vel = 7;
}
if (CurrentTime >= 0.6*1e3 && CurrentTime < 1.6*1e3)
{
ref_vel = 3;
}
/* Sebességszűrés */
ztmp_1=ad11* z_1+ad12* z_2+ad13* z_3 + bd1* ResultData.Velocity;
ztmp_2=ad21* z_1+ad22* z_2+ad23* z_3 + bd2* ResultData.Velocity;
z_3=ad31* z_1+ad32* z_2+ad33* z_3 + bd3* ResultData.Velocity;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
//Sebesség szabályozás hbaszámítása
error_vel=ref_vel- ResultData.Velocity;
error_vel_int = error_vel_int + error_vel*(CurrentTime -
OldTime)/1000;
ResultData.StateVariable_5 = ref_vel;
ResultData.StateVariable_6 = error_vel;
ResultData.StateVariable_7 = error_vel_int;
ResultData.Torque = P_par*error_vel + I_par*error_vel_int - load;
if (ResultData.Torque > 5) { ResultData.Torque = 5; }
if (ResultData.Torque < -5) { ResultData.Torque = -5; }

```

1.7. PI szabályozó terhelésrádobással

Vizsgálja meg a PI szabályozó zavarérzékenységét. A zavart $t = 0.5$ s-ban adjuk a rendszerre, mint negatív nyomatékot $T_{load} = -0.5$, amely kivonódik a szabályozó által számolt kimeneti feszültségből.



7. ábra. PI szabályozó terhelésrádobással

```
// A mérés dátuma és időpontja: 2015. 05. 07. 22:21 CET
// IP Address (IPv4): 86.59.221.241
//Measurement length in milliseconds: 1600
//The state variable names:
//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. ref
//6. error
//7. integral

//declarations:
double P_par = 2.045;
double I_par = 46.55;
double ref_vel = 7;
double error_vel;
static double error_vel_int=0.0;
double load = 6.0;
/* sebesség szűrés változói */
static float z_1=0.0, z_2=0.0, z_3=0.0;
static float ztmp_1=0.0, ztmp_2=0.0;
/* Tsample=1e-3 és Tc=0.0027 */
float ad11= 0.9936, ad12= 9.4621e-004, ad13= 3.4524e-007;
```

```

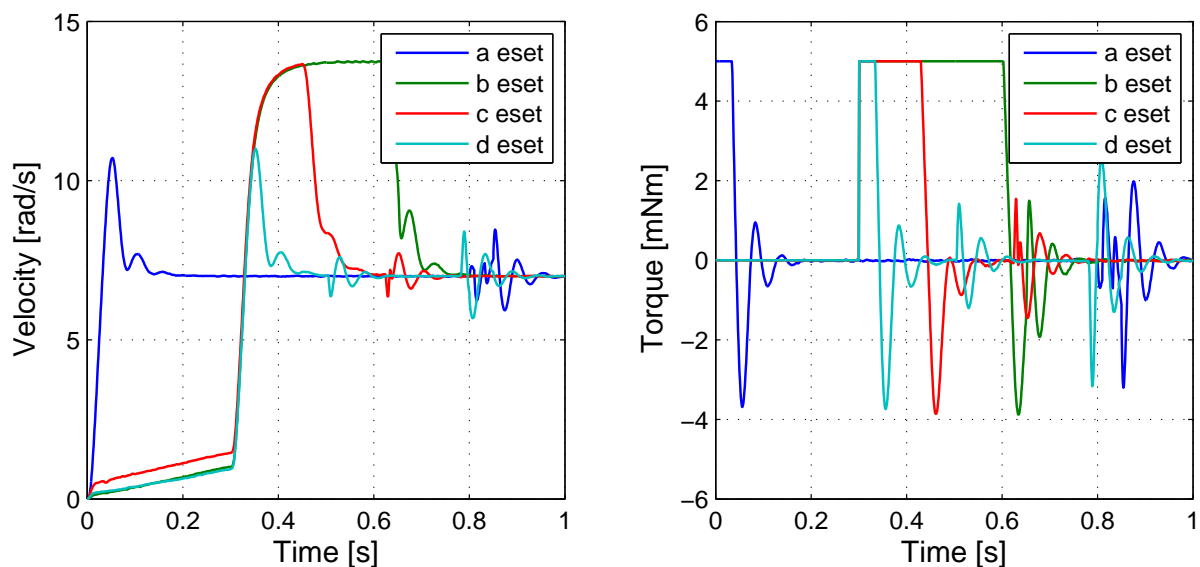
float ad21= - 17.5400, ad22= 0.8515, ad23= 5.6261e-004;
float ad31= -2.8584e+004, ad32= -249.0676, ad33= 0.2264;
float bd1= 0.0064, bd2= 17.5400, bd3= 2.8584e+004;

//controller:
/* Sebesség szűrés */
ztmp_1=ad11* z_1+ad12* z_2+ad13* z_3 + bd1* ResultData.Velocity;
ztmp_2=ad21* z_1+ad22* z_2+ad23* z_3 + bd2* ResultData.Velocity;
z_3=ad31* z_1+ad32* z_2+ad33* z_3 + bd3* ResultData.Velocity;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
if (CurrentTime > 0*1e3 && CurrentTime < 0.3*1e3)
{
load = 2.0;
}
if (CurrentTime >= 0.3*1e3 && CurrentTime < 0.7*1e3)
{
load = 0.5;
}
if (CurrentTime >= 0.7*1e3 && CurrentTime < 1.6*1e3)
{
load = 2.0;
}
//Sebesség szabályozó hibaszámítása
error_vel=ref_vel- ResultData.Velocity;
error_vel_int = error_vel_int + error_vel*(CurrentTime - OldTime)/1000.0;
ResultData.StateVariable_5 = ref_vel;
ResultData.StateVariable_6 = error_vel;
ResultData.StateVariable_7 = error_vel_int;
ResultData.Torque = P_par*error_vel + I_par*error_vel_int - load;
if (ResultData.Torque > 5)
{
ResultData.Torque = 5;
}
if (ResultData.Torque < -5)
{
ResultData.Torque = -5;
}

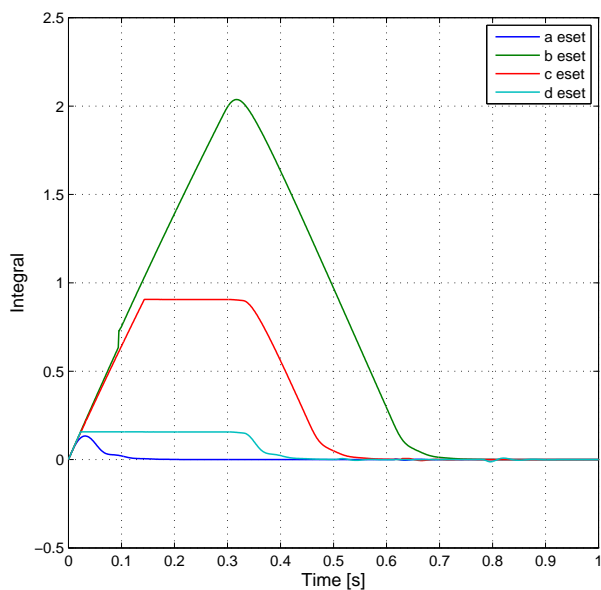
```


1.8. Anti windup PI szabályozó

A PI szabályozó hibatoleranciája (maradó hiba mentes PI szabályozó) Szimuláljunk hibát, amikor a motor megakad, vagy az elektromos egység meghibásodik és a nyomaték 0, miközben a PI szabályozó működik. Az integrál tag növekszik. A hiba után az integráló tag értékét le kell csökkenteni (a pozitív hiba közben integráló tag értéke nagyon megnövekedhet, amelyet egy negatív hibával lehet csökkenteni).



8. ábra. PI szabályozó hibatoleranciája



9. ábra. PI szabályozó hibatoleranciája, Integrálás

```

// A mérés dátuma és időpontja: 2015. 05. 09. 23:23 CET
// IP Address (IPv4): 86.59.221.130
//Measurement length in milliseconds: 12000
//The state variable names:
//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. ref
//6. error
//7. integral

//declarations:
double P_par = 2;
double I_par = 50;
double ref_vel = 7;
double error_vel;
static double error_vel_int=0.0;
double load = 0;
/*
// a esetben
double int_lim = 100;
double Tf = -0.3*1e3;
// b esetben
double int_lim = 100;
double Tf = 0.3*1e3;
// c esetben
double int_lim = 0.9;
double Tf = 0.3*1e3;
// d esetben
double int_lim = 0.15;
double Tf = 0.3*1e3;
*/
// kérem másolja be az egyik pár paramétert
double int_lim = 0.15;
double Tf = 0.3*1e3;
/* sebesség szűrő változói */
static float z_1=0.0, z_2=0.0, z_3=0.0;
static float ztmp_1=0.0, ztmp_2=0.0;
/* Tsample=1e-3 és Tc=0.0027 */
float ad11= 0.9936, ad12= 9.4621e-004, ad13= 3.4524e-007;
float ad21= - 17.5400, ad22= 0.8515, ad23= 5.6261e-004;
float ad31= -2.8584e+004, ad32= -249.0676, ad33= 0.2264;
float bd1= 0.0064, bd2= 17.5400, bd3= 2.8584e+004;

//controller:
/* Sebesség szűrő */
ztmp_1=ad11* z_1+ad12* z_2+ad13* z_3 + bd1* ResultData.Velocity;
ztmp_2=ad21* z_1+ad22* z_2+ad23* z_3 + bd2* ResultData.Velocity;
z_3=ad31* z_1+ad32* z_2+ad33* z_3 + bd3* ResultData.Velocity;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
//Sebesség szabályozó hibaszámítása
error_vel=ref_vel- ResultData.Velocity;

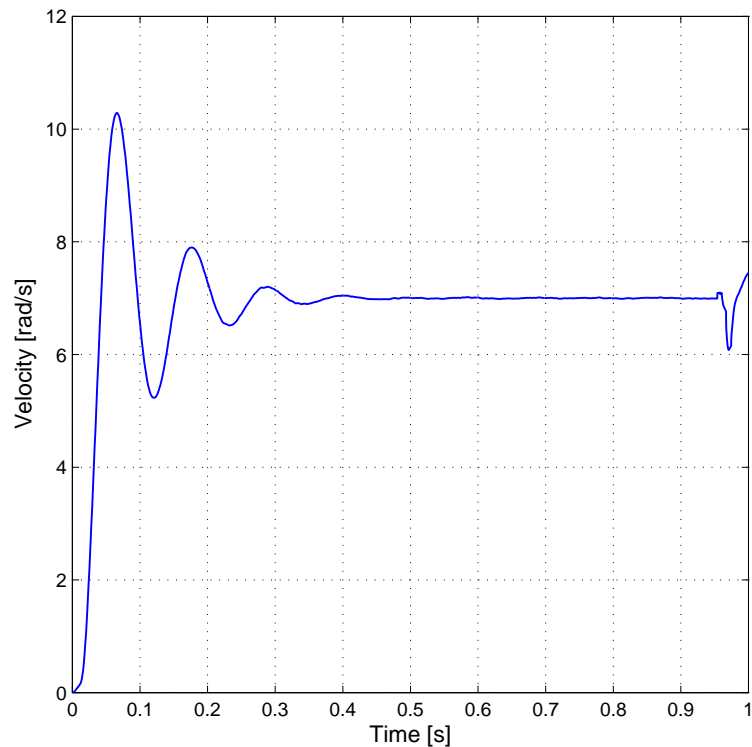
```

```

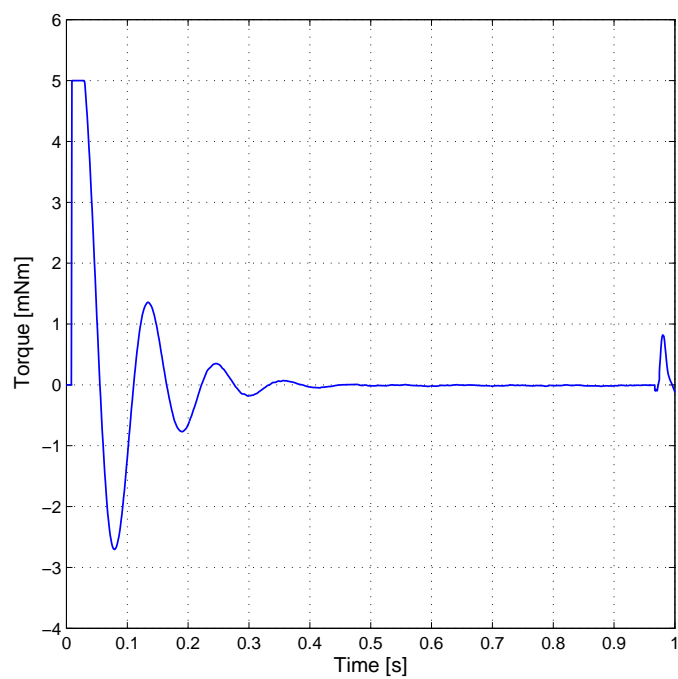
error_vel_int = error_vel_int + error_vel*(CurrentTime -
OldTime)/1000.0;
ResultData.StateVariable_5 = ref_vel;
ResultData.StateVariable_6 = error_vel;
ResultData.StateVariable_7 = error_vel_int;
if (error_vel_int > int_lim)
{
error_vel_int = int_lim;
}
ResultData.Torque = P_par*error_vel + I_par*error_vel_int - load;
if ( CurrentTime < Tf)
{
ResultData.Torque = 0.0;
}
if (ResultData.Torque > 5)
{
ResultData.Torque = 5;
}
if (ResultData.Torque < -5)
{
ResultData.Torque = -5;
}

```

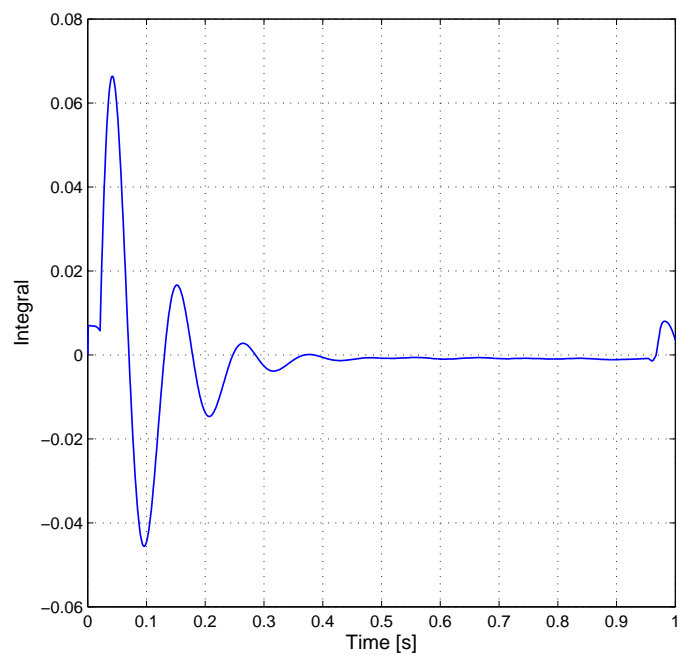
1.9. Időkésleltetett rendszer PI szabályozása



10. ábra. Időkésleltetett PI szabályozó, Sebesség



11. ábra. Időkésleltetett PI szabályozó, Nyomaték



12. ábra. Időkésleltetett PI szabályozó, Integrálás

```
// A mérés dátuma és időpontja: 2015. 05. 09. 23:30 CET
// IP Address (IPv4): 86.59.221.130
//Measurement length in milliseconds: 1000
//The state variable names:
```

```

//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. ref
//6. error
//7. integral

//declarations:
double P_par = 0.85;
double I_par = 12.0;
double ref_vel = 7;
double error_vel;
static double error_vel_int=0.0;
double load = 0;
static float T_1=0.0, T_2=0.0, T_3=0.0;
static float T_4=0.0, T_5=0.0, T_6=0.0;
static float T_7=0.0, T_8=0.0, T_9=0.0;
/* velocity filter variables */
static float z_1=0.0, z_2=0.0, z_3=0.0;
static float ztmp_1=0.0, ztmp_2=0.0;
/* Tsample=1e-3 and Tc=0.0032 */
float ad11 = 0.99591, ad12 = 0.00095987, ad13 = 3.652e-007;
float ad21 = -11.3235, ad22 = 0.88778, ad23 = 0.00061567;
float ad31 = -19089.6748, ad32 = -193.6165, ad33 = 0.30752;
float bd1 = 0.0040906, bd2 = 11.3235, bd3 = 19089.6748;

//controller:
/* Velocity filter */
ztmp_1=ad11* z_1+ad12* z_2+ad13* z_3 + bd1* ResultData.Velocity;
ztmp_2=ad21* z_1+ad22* z_2+ad23* z_3 + bd2* ResultData.Velocity;
z_3=ad31* z_1+ad32* z_2+ad33* z_3 + bd3* ResultData.Velocity;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
//Error calculation for velocity control
error_vel=ref_vel- ResultData.Velocity;
error_vel_int = error_vel_int + error_vel*(CurrentTime -
OldTime)/1000;
ResultData.StateVariable_5 =ref_vel;
ResultData.StateVariable_6 = error_vel;
ResultData.StateVariable_7 = error_vel_int;
ResultData.Torque = T_1;
T_1=T_2;
T_2=T_3;
T_3=T_4;
T_4=T_5;
T_5=T_6;
T_6=T_7;
T_7=T_8;
T_8=T_9;
T_9= P_par*error_vel + I_par*error_vel_int - load;
if (T_9 > 5) { T_9 = 5;
error_vel_int = error_vel_int - error_vel*(CurrentTime -
OldTime)/1000;}

```

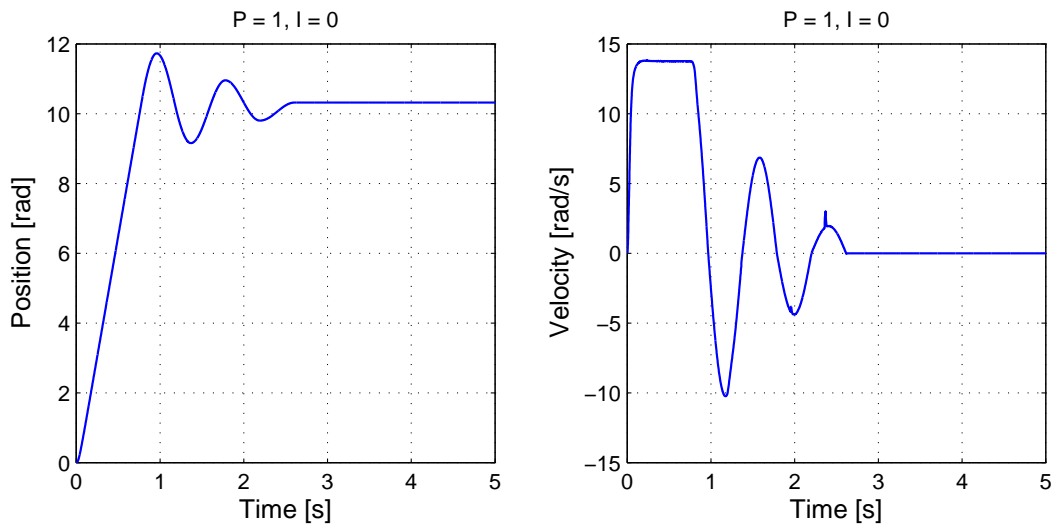
```

if (T_9 < -5) { T_9 = -5;
error_vel_int = error_vel_int - error_vel*(CurrentTime -
OldTime)/1000;}

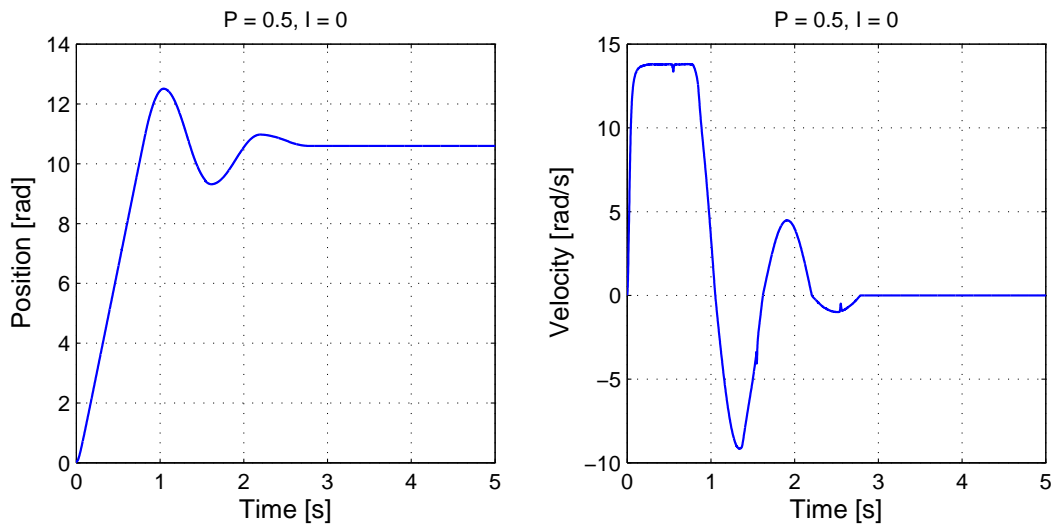
```

1.10. P és PI szabályozó ugrás függvény pozíció referencijel válasza

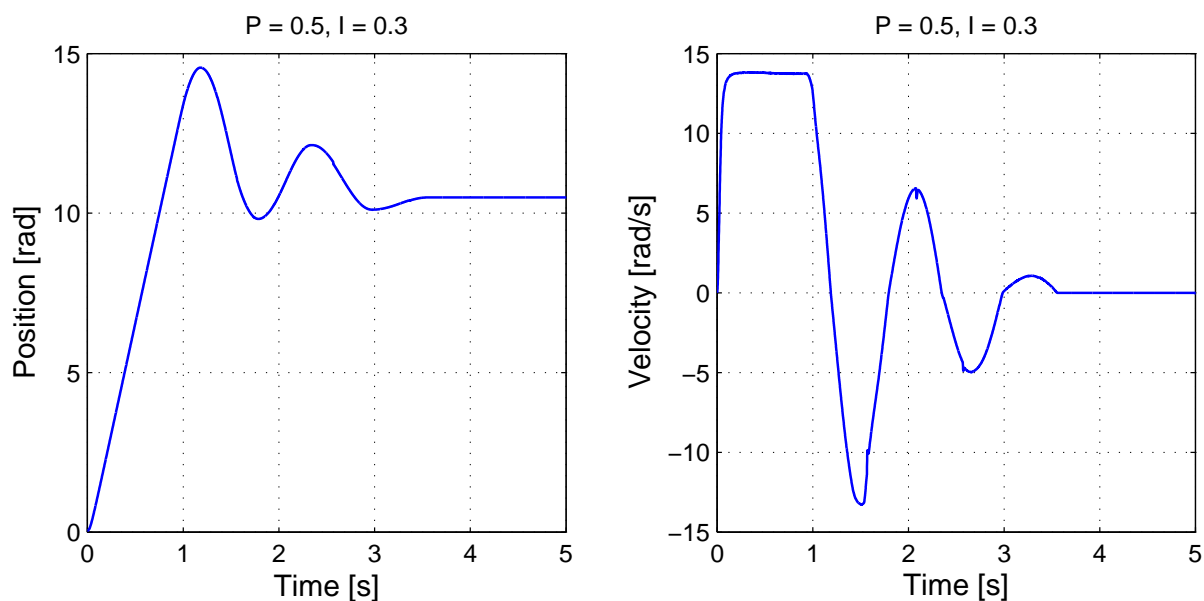
Ehhez a teszthez a referencia tengely-pozíciót 10 rad-ra állítjuk.



13. ábra. P szabályozó a referencia pozíció ugrásszerű változására, $P = 1$, $I = 0$



14. ábra. P szabályozó a referencia pozíció ugrásszerű változására, $P = 0.5$, $I = 0$



15. ábra. P szabályozó a referencia pozíció ugrásszerű változására, $P = 0.5$, $I = 0.3$

```
// A mérés dátuma és időpontja: 2015. 05. 09. 23:39 CET
// IP Address (IPv4): 86.59.221.130
//Measurement length in milliseconds: 5000
//The state variable names:
//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. ref
//6. error
//7. integral

//declarations:
double P_par = 1;
double I_par = 0;
double ref_pos = 10;
double error_pos;
static double error_pos_int=0.0;
double load = 0;
/* sebesség szűrés változói */
static float z_1=0.0, z_2=0.0, z_3=0.0;
static float ztmp_1=0.0, ztmp_2=0.0;
/* Tsample=1e-3 és Tc=0.0027 */
float ad11= 0.9936, ad12= 9.4621e-004, ad13= 3.4524e-007;
float ad21= -17.5400, ad22= 0.8515, ad23= 5.6261e-004;
float ad31= -2.8584e+004, ad32= -249.0676, ad33= 0.2264;
float bd1= 0.0064, bd2= 17.5400, bd3= 2.8584e+004;
static double ini_0 = 0;
static double ini_1 = -10;

//controller:
```

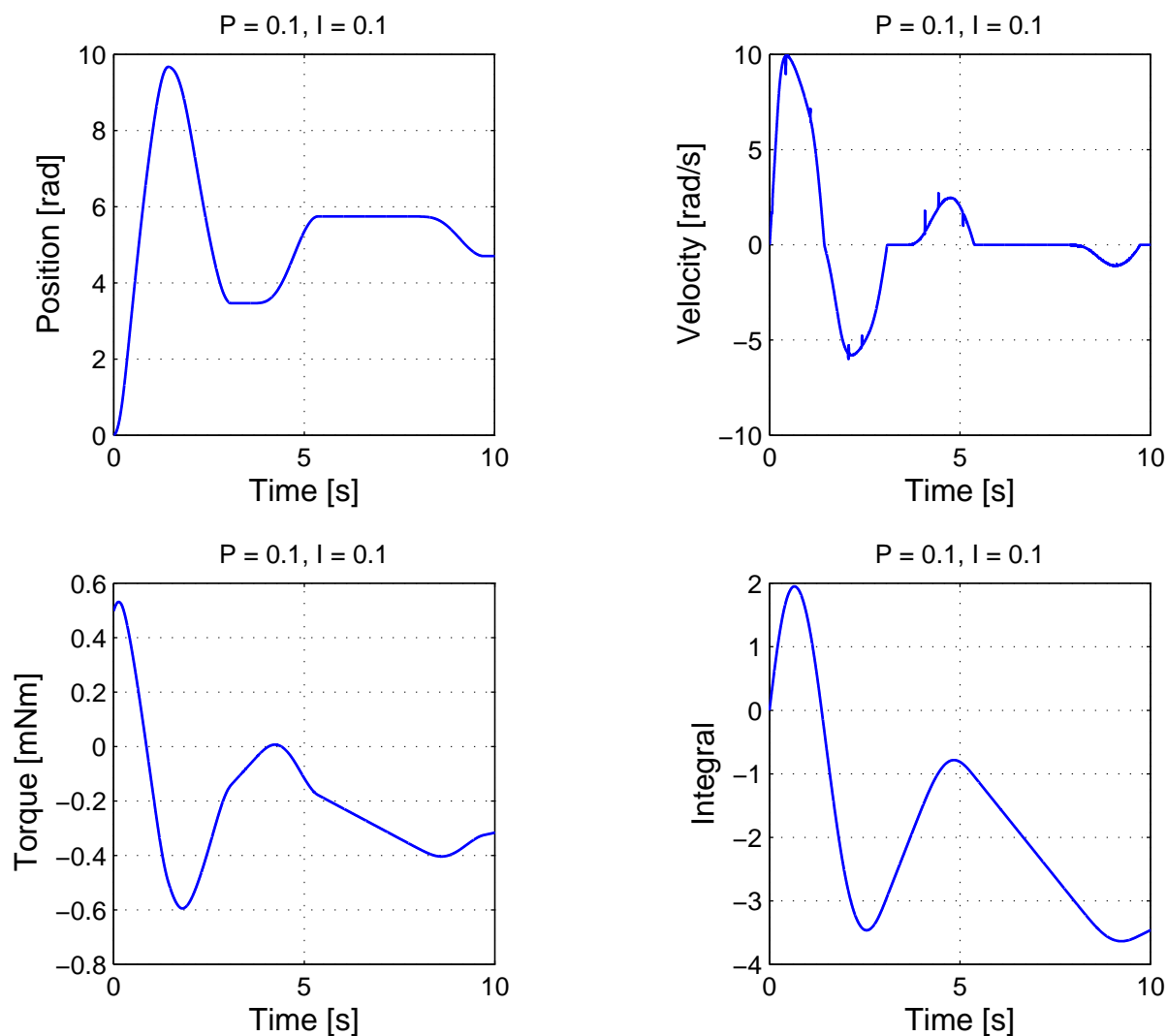
```

if (ini_1 < 0)
{
ini_0 = ResultData.Position;
}
ini_1 = 5;
/* Velocity filter */
ztmp_1=ad11* z_1+ad12* z_2+ad13* z_3 + bd1* ResultData.Velocity;
ztmp_2=ad21* z_1+ad22* z_2+ad23* z_3 + bd2* ResultData.Velocity;
z_3=ad31* z_1+ad32* z_2+ad33* z_3 + bd3* ResultData.Velocity;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
//Sebesség szabályozó hibaszámítása
error_pos=ref_pos- ResultData.Position + ini_0;
error_pos_int = error_pos_int + error_pos*(CurrentTime -
OldTime)/1000.0;
ResultData.StateVariable_5 = ref_pos;
ResultData.StateVariable_6 = error_pos;
ResultData.StateVariable_7 = error_pos_int;
ResultData.StateVariable_8 = ResultData.Position - ini_0;
ResultData.Torque = P_par*error_pos + I_par*error_pos_int - load;
if (ResultData.Torque > 5)
{
ResultData.Torque = 5;
}
if (ResultData.Torque < -5)
{
ResultData.Torque = -5;
}

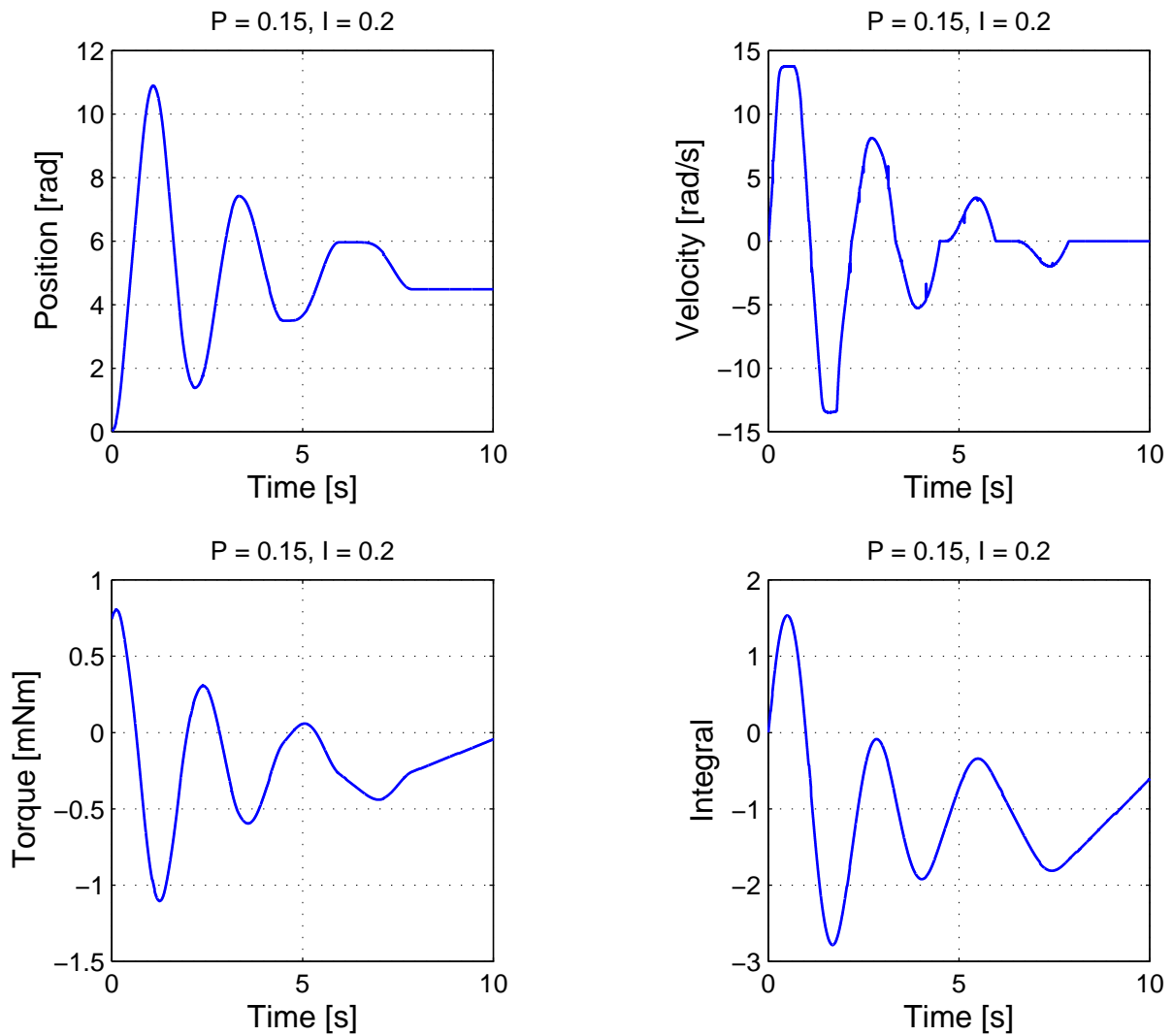
```


1.11. Stick-slip jelenség

Válassza meg a P és I paramétert úgy, hogy a motor ragadjon be a referencia pozíció előtt, és az integrátor lendítse túl a hibán.



16. ábra. Stick-slip jelenség, $P = 0.1$, $I = 0.1$



17. ábra. Stick-slip jelenség, $P = 0.15$, $I = 0.2$

```
// A mérés dátuma és időpontja: 2015. 05. 09. 23:45 CET
// IP Address (IPv4): 86.59.221.130
//Measurement length in milliseconds: 10000
//The state variable names:
//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. ref
//6. error
//7. integral

//declarations:
double P_par = 0.1;
double I_par = 0.1;
```

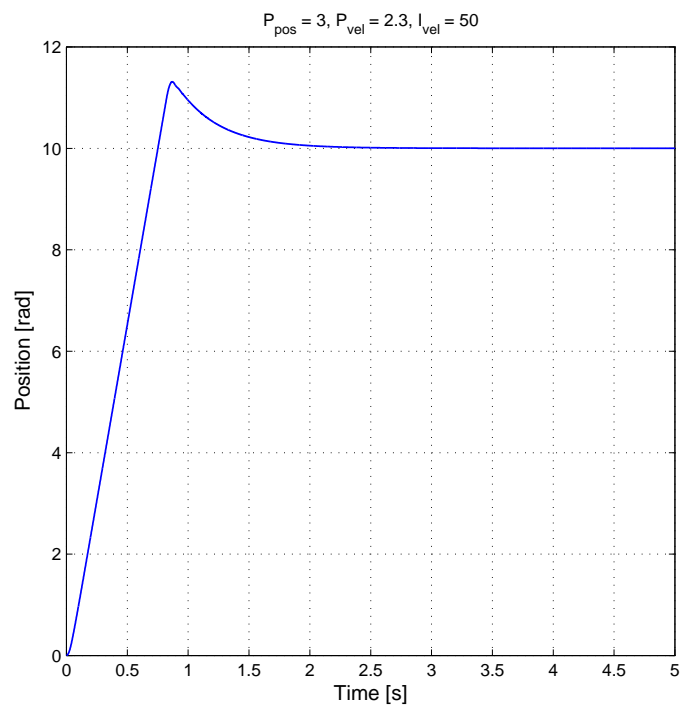
```

double ref_pos = 5;
double error_pos;
static double error_pos_int=0.0;
double load = 0;
/* sebesség szűrés változói */
static float z_1=0.0, z_2=0.0, z_3=0.0;
static float ztmp_1=0.0, ztmp_2=0.0;
/* Tsample=1e-3 és Tc=0.0027 */
float ad11= 0.9936, ad12= 9.4621e-004, ad13= 3.4524e-007;
float ad21= - 17.5400, ad22= 0.8515, ad23= 5.6261e-004;
float ad31= -2.8584e+004, ad32= -249.0676, ad33= 0.2264;
float bd1= 0.0064, bd2= 17.5400, bd3= 2.8584e+004;
static double ini_0 = 0;
static double ini_1 = -10;

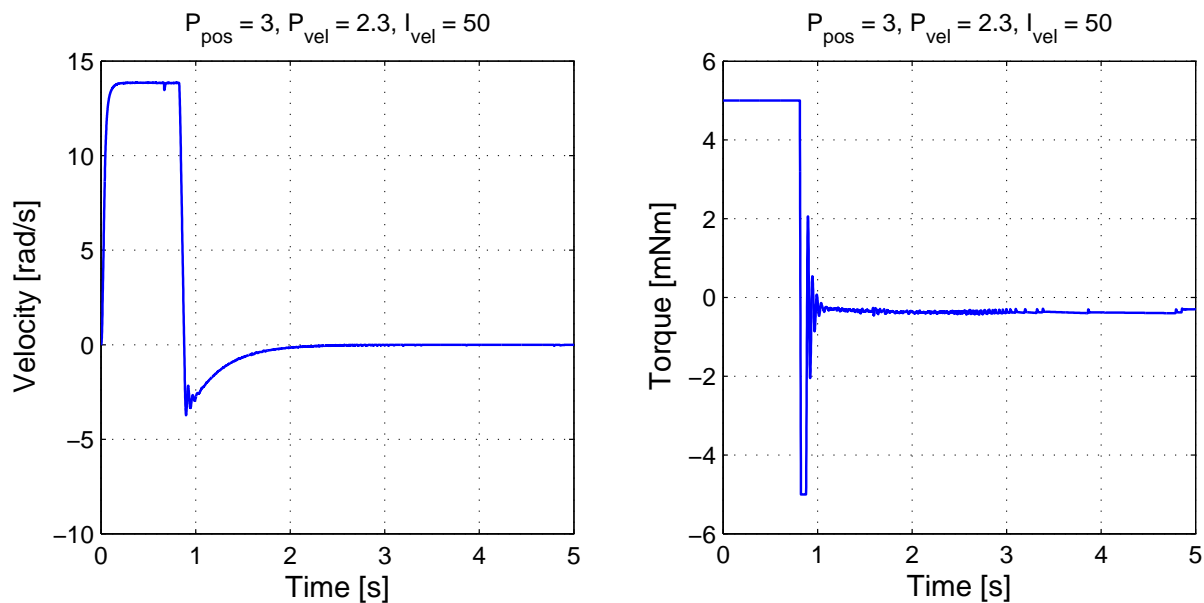
//controller:
if (ini_1 < 0)
{
ini_0 = ResultData.Position;
}
ini_1 =5;
/* Sebesség szűrés */
ztmp_1=ad11* z_1+ad12* z_2+ad13* z_3 + bd1* ResultData.Velocity;
ztmp_2=ad21* z_1+ad22* z_2+ad23* z_3 + bd2* ResultData.Velocity;
z_3=ad31* z_1+ad32* z_2+ad33* z_3 + bd3* ResultData.Velocity;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
//Sebesség szabályozó hibaszámítása
error_pos=ref_pos- CurrentPosition;
error_pos_int = error_pos_int + error_pos*(CurrentTime -
OldTime)/1000.0;
ResultData.StateVariable_5 = ref_pos;
ResultData.StateVariable_6 = error_pos;
ResultData.StateVariable_7 = error_pos_int;
ResultData.Torque = P_par*error_pos + I_par*error_pos_int - load;
if (ResultData.Torque > 5)
{
ResultData.Torque = 5;
}
if (ResultData.Torque < -5)
{
ResultData.Torque = -5;
}

```

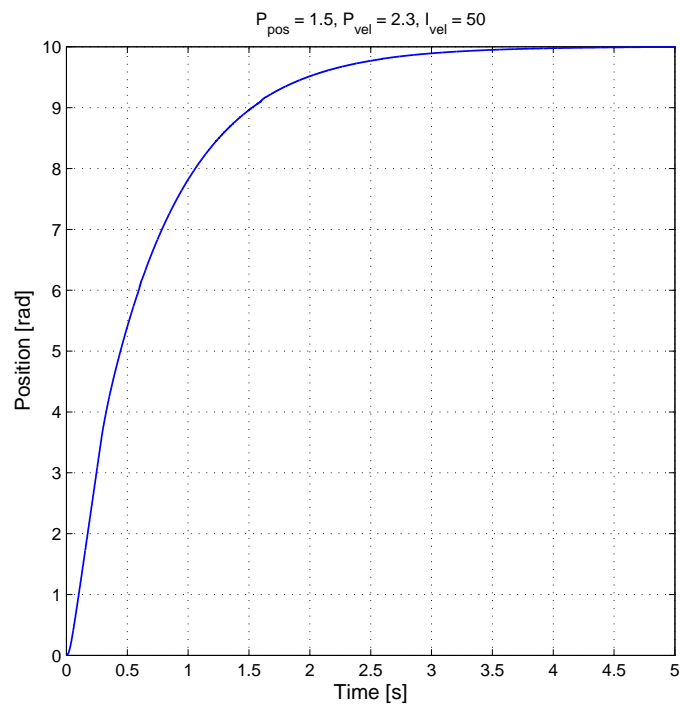
1.12. Pozíció szabályozás belső sebesség szabályozóval



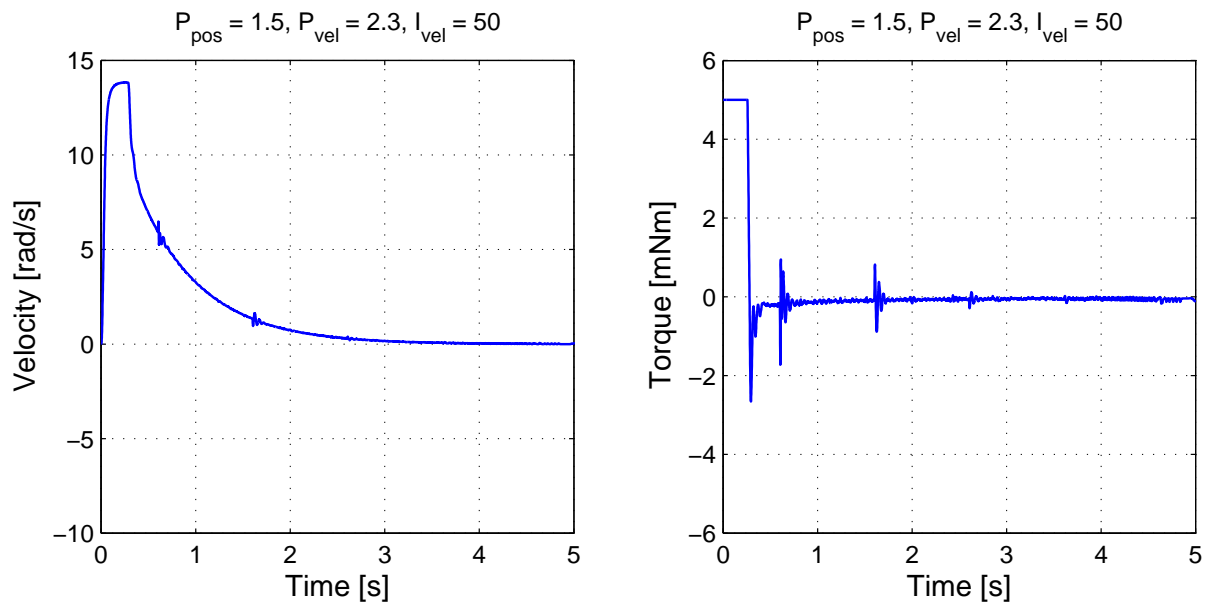
18. ábra. Kaszkád szabályozás, Pozíció $P_{pos} = 3, P_{vel} = 2.3, I_{vel} = 50$



19. ábra. Kaszkád szabályozás, $P_{pos} = 3, P_{vel} = 2.3, I_{vel} = 50$



20. ábra. Kaszkád szabályozás, Pozíció $P_{pos} = 1.5, P_{vel} = 2.3, I_{vel} = 50$



21. ábra. Kaszkád szabályozás, $P_{pos} = 1.5, P_{vel} = 2.3, I_{vel} = 50$

```
// A mérés dátuma és időpontja: 2015. 05. 10. 00:02 CET
// IP Address (IPv4): 86.59.221.130
//Measurement length in milliseconds: 5000
//The state variable names:
```

```

//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. ref_poz (választott)
//6. ref_vel (választott)
//7. err_pos (választott)
//8. err_vel (választott)
//9. int_vel (választott)
//10. poz (választott)

//declarations:
double P_pos = 3;
double P_vel = 2.3;
double I_vel = 50;
double ref_pos = 10;
double ref_vel = 0;
double error_pos = 0;
double error_vel = 0;
static double error_vel_int = 0;
static double ini_0 = 0;
static double ini_1 = -10;
/* sebesség szabályozó változói */
static float z_1=0.0, z_2=0.0, z_3=0.0;
static float ztmp_1=0.0, ztmp_2=0.0;
/* Tsamplé=1e-3 és Tc=0.0027 */
float ad11= 0.9936, ad12= 9.4621e-004, ad13= 3.4524e-007;
float ad21= - 17.5400, ad22= 0.8515, ad23= 5.6261e-004;
float ad31= -2.8584e+004, ad32= -249.0676, ad33= 0.2264;
float bd1= 0.0064, bd2= 17.5400, bd3= 2.8584e+004;

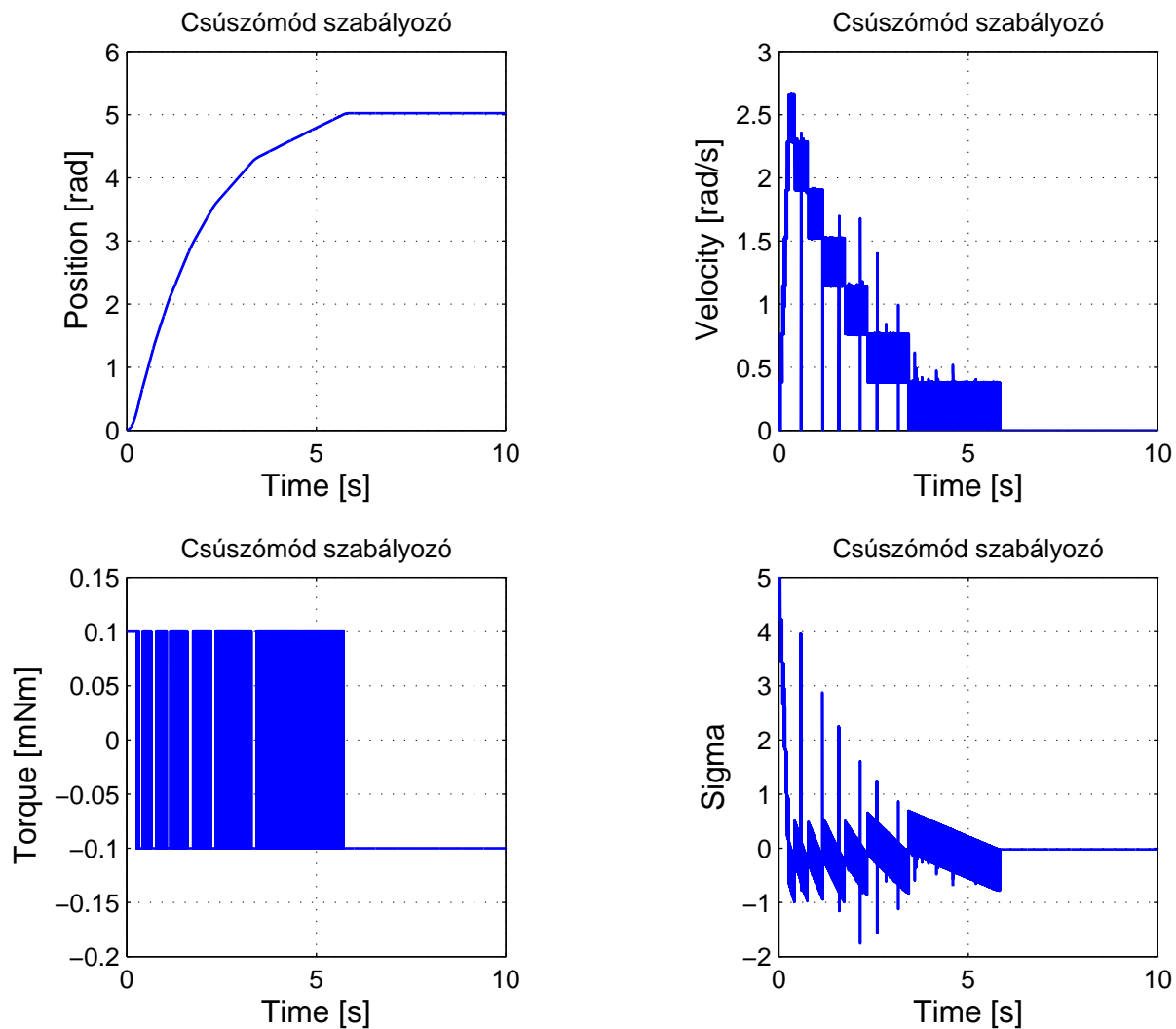
//controller:
if (ini_1 < 0)
{
ini_0 = ResultData.Position;
}
ini_1 = 5;
/* Sebesség szabályozó */
ztmp_1=ad11* z_1+ad12* z_2+ad13* z_3 + bd1* ResultData.Velocity;
ztmp_2=ad21* z_1+ad22* z_2+ad23* z_3 + bd2* ResultData.Velocity;
z_3=ad31* z_1+ad32* z_2+ad33* z_3 + bd3* ResultData.Velocity;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
//Pozíció szabályozás hibaszámítása
error_pos = ref_pos- ResultData.Position + ini_0;
// Sebesség szabályozás hibaszámítása
ref_vel = error_pos*P_pos;
error_vel = ref_vel - ResultData.Velocity;
error_vel_int = error_vel_int + error_vel*(CurrentTime -
OldTime)/1000;
ResultData.StateVariable_5 = ref_pos;
ResultData.StateVariable_6 = ref_vel;
ResultData.StateVariable_7 = error_pos;
ResultData.StateVariable_8 = error_vel;

```

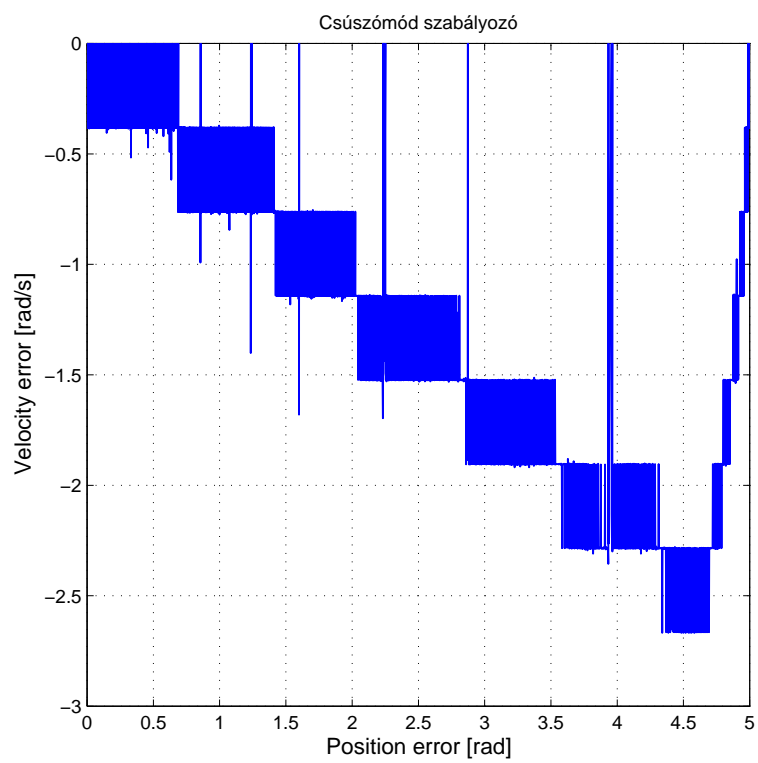
```
ResultData.StateVariable_9 = error_vel_int;  
ResultData.StateVariable_10 = ResultData.Position - ini_0;  
//Szabályozó  
ResultData.Torque = P_vel*error_vel + I_vel*error_vel_int;  
if (ResultData.Torque > 5)  
{  
ResultData.Torque = 5;  
}  
if (ResultData.Torque < -5)  
{  
ResultData.Torque = -5;  
}
```

1.13. Csúszómód szabályozó

Tervezzen csúszómód szabályozót a motor pozíciójához.



22. ábra. A pozíció szabályozó eredményei csúszómódban



23. ábra. A pozíció szabályozó eredményei csúszómódban, Hiba

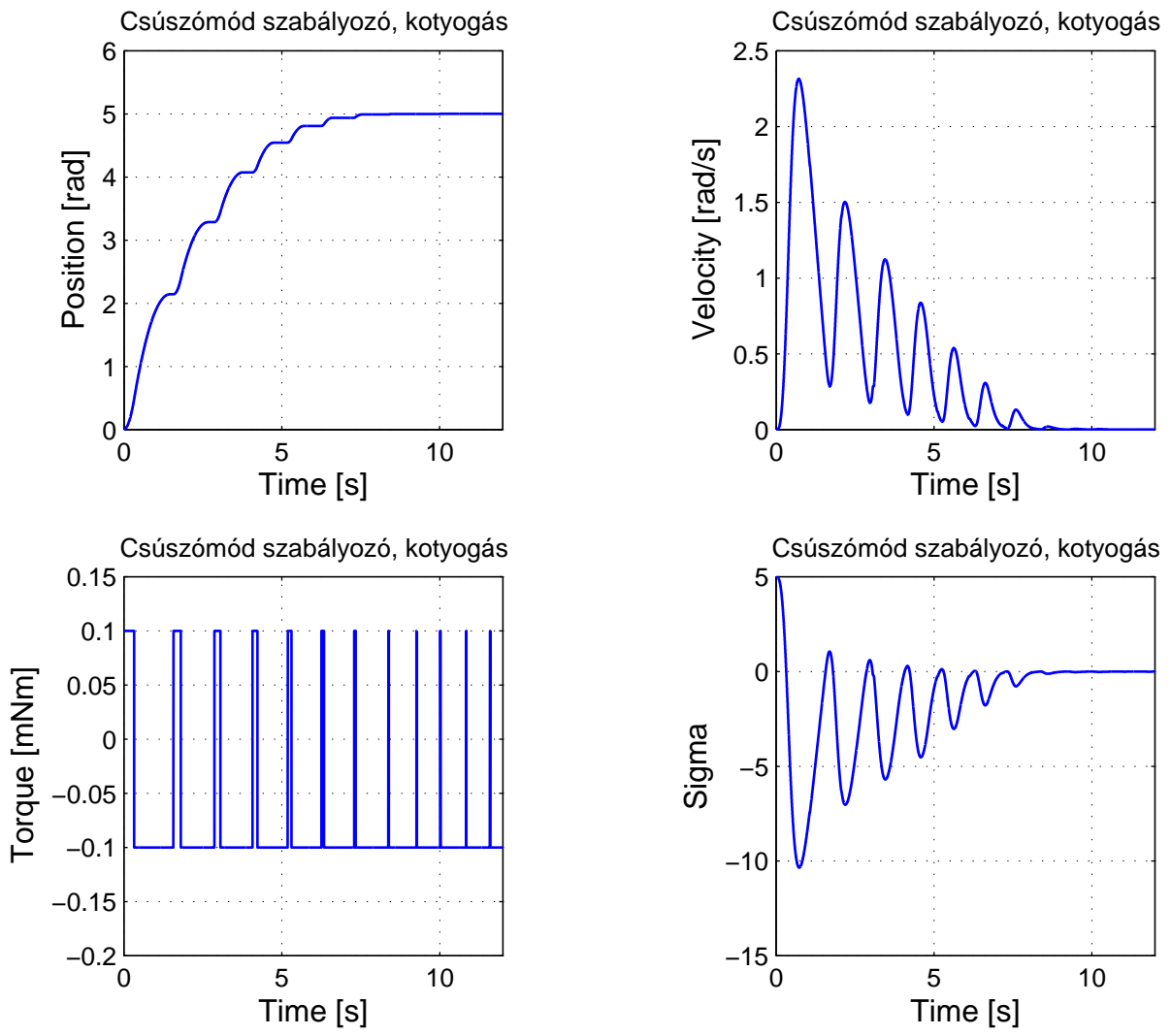
```

// A mérés dátuma és időpontja: 2015. 05. 10. 00:15 CET
// IP Address (IPv4): 86.59.221.130
//Measurement length in milliseconds: 12000
//The state variable names:
//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. sigma (választott)

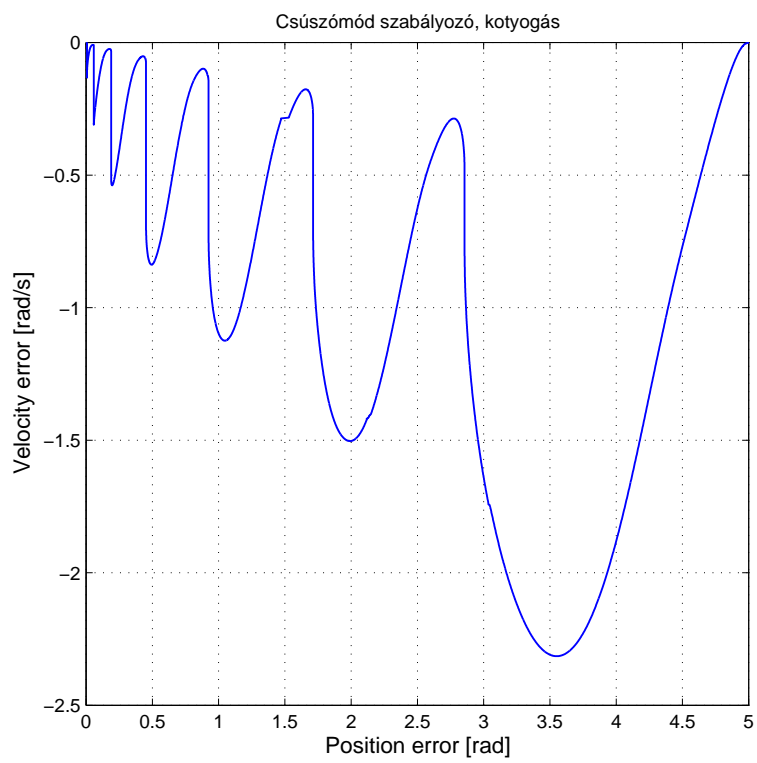
//declarations:
float sigma;
float error;
float error_dot;
float ref=5.0;
float lambda=2;
static double ini_0 = 0;
static double ini_1 = -10;

//controller:
if (ini_1 < 0)
{
ini_0 = ResultData.Position;
}
ini_1 = 5;
error=ref-ResultData.Position+ ini_0;
error_dot=- ResultData.Velocity;
sigma= error+ lambda*error_dot;
ResultData.StateVariable_5 = sigma;
if (sigma>0)
{ ResultData.Torque=0.1;
}
if (sigma<0)
{ ResultData.Torque =-0.1;
}
if (sigma=0)
{ ResultData.Torque=0;
}

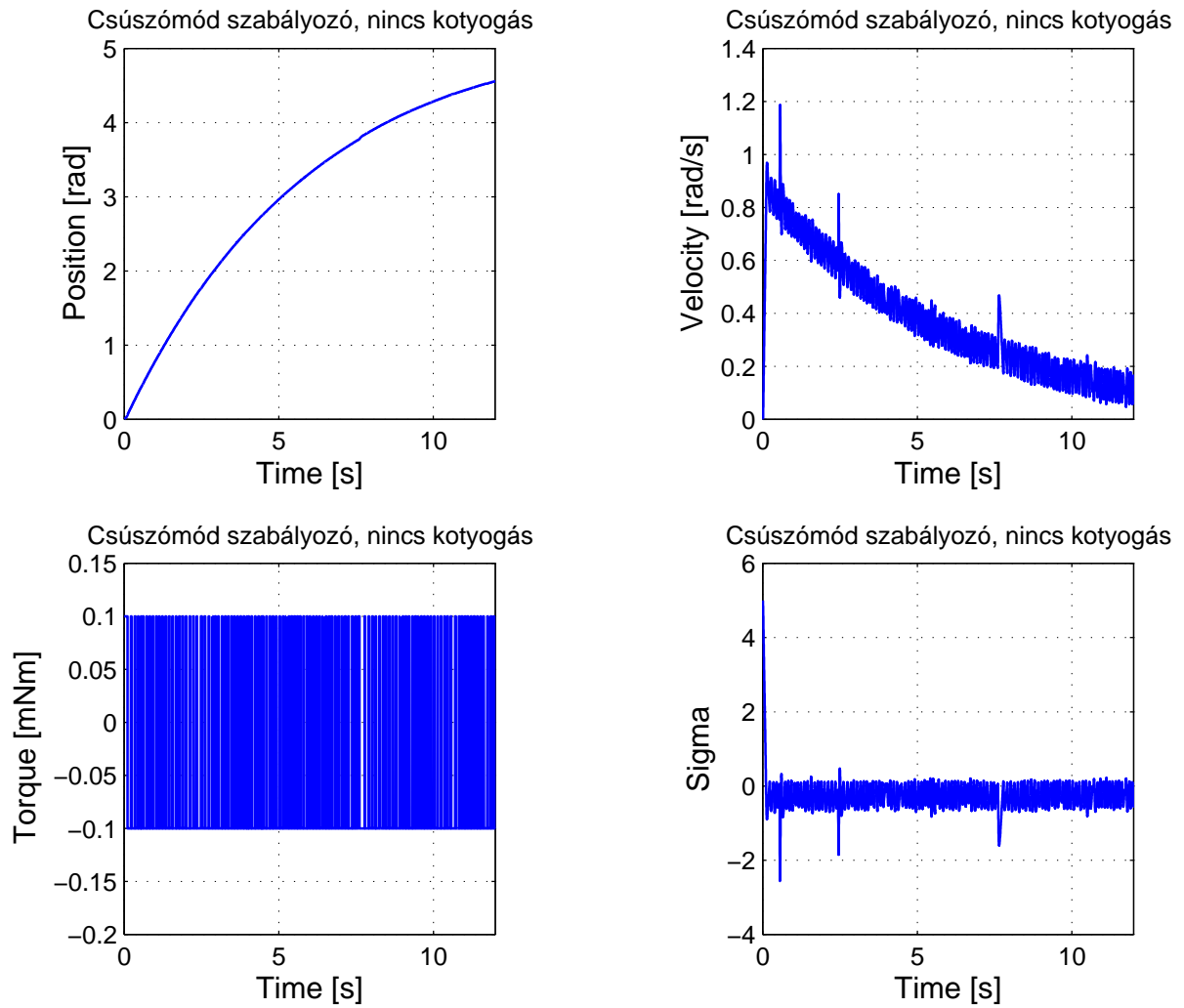
```



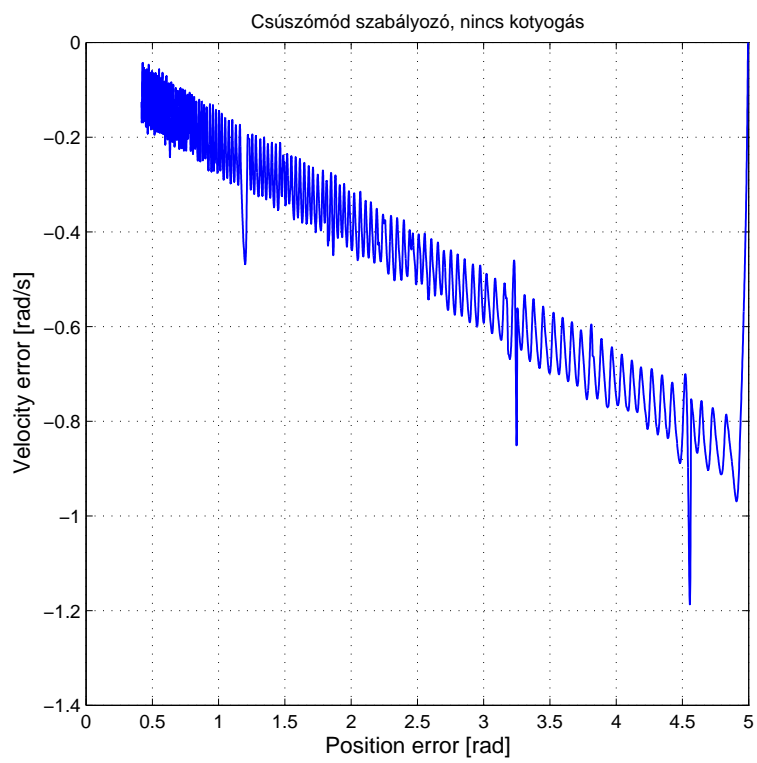
24. ábra. Csúszómód szabályozó sebesség szűrővel (kotyogás)



25. ábra. Csúszómód szabályozó sebesség szűrővel (kotyogás), Hiba



26. ábra. Csúszómód szabályozó sebesség szűrővel (nincs kotyogás)



27. ábra. Csúszómód szabályozó sebesség szűrővel (nincs kotyogás), Hiba

```

// A mérés dátuma és időpontja: 2015. 05. 10. 00:28 CET
// IP Address (IPv4): 86.59.221.130
//Measurement length in milliseconds: 12000
//The state variable names:
//1. time (given)
//2. position (given)
//3. velocity (given)
//4. torque (given)
//5. sigma (választott)

//declarations:
float sigma;
float error;
float error_dot;
float ref=5.0;
float lambda=6;
/* filter variables*/
static float z_1=0.0, z_2=0.0, z_3=0.0;
static float ztmp_1=0.0, ztmp_2=0.0;
/*omega_c=10 a modellezetlen dinamika nagy ketyogást okoz */
float Azd11= 1, Azd12= 0.0010, Azd13= 0.0000;
float Azd21= -0.0005, Azd22= 0.9999, Azd23= 0.0010;
float Azd31= -0.9851, Azd32= -0.2960, Azd33= 0.9703;
float Bzd1= 0.0000, Bzd2= 0.0005, Bzd3= 0.9851;
/* omega_c=1/0.007 a modellezetlen dinamika nem okoz nagy ketyogást
float Azd11= 0.9996, Azd12= 9.9072e-004, Azd13= 4.3344e-007;
float Azd21= -1.2637, Azd22= 0.9730, Azd23= 8.0496e-004;
float Azd31= -2.3468e+003, Azd32= -50.5468, Azd33= 0.6280;
float Bzd1= 4.3671e-004, Bzd2= 1.2637, Bzd3= 2.3468e+003;
*/
static double ini_0 = 0;
static double ini_1 = -10;

//controller:
if (ini_1 < 0)
{
ini_0 = ResultData.Position;
}
ini_1 = 5;
error=ref-ResultData.Position+ ini_0;
/* filter */
ztmp_1 = Azd11* z_1 +Azd12* z_2 +Azd13* z_3 +
Bzd1*ResultData.Velocity ;
ztmp_2 = Azd21* z_1 +Azd22* z_2 +Azd23* z_3 +
Bzd2*ResultData.Velocity ;
z_3 = Azd31*z_1 +Azd32*z_2 +Azd33* z_3 + Bzd3*ResultData.Velocity ;
z_1 = ztmp_1;
z_2 = ztmp_2;
ResultData.Velocity =z_1;
error_dot=- z_1;
sigma=error+ lambda*error_dot;
ResultData.StateVariable_5 = sigma;
if (sigma>0)
{ ResultData.Torque=0.1;}
if (sigma<0)

```

```
{ ResultData.Torque =-0.1;  
}  
if (sigma=0)  
{ ResultData.Torque=0;}
```


2. Melléklet

Ábrák Matlab kódja

```
close all
clear all
current_dir = cd;

cd('Exercise-5\Task1-P-hangolas');
Time      % t vector
Torque    % torque vector
Velocity  % velocity vector

Plot_Task1 = figure('Position',[50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
xlabel('Time [s]','FontSize',12)
ylabel('Velocity [rad/s]','FontSize',12)
xlim([0 1]);
ylim([-1.5 1.5]);
axis square
grid on

subplot(1,2,2)
plot( (time./1000), torque, 'b', 'LineWidth', 1)
xlabel('Time [s]','FontSize',12)
ylabel('Torque [mNm]','FontSize',12)
xlim([0 1]);
ylim([-5.9 5.9]);
axis square
grid on

cd(current_dir);
print(Plot_Task1, 'latex-meres2\fig\task1', '-depsc')

%% Task2-P-ugras-valasz
cd('Exercise-5\Task2-P-ugras-valasz');
Time      % t vector
Torque    % torque vector
Velocity  % velocity vector

Plot_Task2 = figure('Position',[50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
xlabel('Time [s]','FontSize',12)
ylabel('Velocity [rad/s]','FontSize',12)
xlim([0 0.3]);
ylim([-1.5 1.5]);
axis square
grid on
```

```

subplot(1,2,2)
plot( (time./1000), torque, 'b', 'LineWidth', 1)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Torque [mNm]', 'FontSize', 12)
xlim([0 0.3]);
ylim([-4.0 6.0]);
axis square
grid on

cd(current_dir);
print(Plot_Task2, 'latex-meres2\fig\task2', '-depsc')

%% Task3-P-valt-ref
cd('Exercise-5\Task3-P-valt-ref');
Time      % t vector
Torque    % torque vector
Velocity  % velocity vector
Ref_signal % ref vector

Plot_Task3 = figure('Position', [50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), velocity, (time./1000), ref, 'LineWidth', 1)
legend('Velocity', 'Ref signal');
xlabel('Time [s]', 'FontSize', 12)
ylabel('Velocity [rad/s]', 'FontSize', 12)
xlim([0 1]);
%ylim([0 10]);
axis square
grid on

subplot(1,2,2)
plot( (time./1000), torque, 'b', 'LineWidth', 1)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Torque [mNm]', 'FontSize', 12)
xlim([0 1]);
%ylim([-6.0 6.0]);
axis square
grid on

cd(current_dir);
print(Plot_Task3, 'latex-meres2\fig\task3', '-depsc')

%% Task4-P-terheles-ugras
cd('Exercise-5\Task4-P-terheles-ugras');
Time      % t vector
Torque    % torque vector
Velocity  % velocity vector

Plot_Task4 = figure('Position', [50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)

```

```

xlabel('Time [s]', 'FontSize', 12)
ylabel('Velocity [rad/s]', 'FontSize', 12)
%xlim([0 1]);
%ylim([0 10]);
axis square
grid on

subplot(1,2,2)
plot( (time./1000), torque, 'b', 'LineWidth', 1)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Torque [mNm]', 'FontSize', 12)
%xlim([0 1]);
ylim([-4.0 6.0]);
axis square
grid on

cd(current_dir);
print(Plot_Task4, 'latex-meres2\fig\task4', '-depsc')

%% Task5-PI-ugras-valasz
cd('Exercise-5\Task5-PI-ugras-valasz');
Time      % t vector
Torque    % torque vector
Velocity  % velocity vector

Plot_Task5 = figure('Position', [50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Velocity [rad/s]', 'FontSize', 12)
%xlim([0 1]);
%ylim([0 10]);
axis square
grid on

subplot(1,2,2)
plot( (time./1000), torque, 'b', 'LineWidth', 1)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Torque [mNm]', 'FontSize', 12)
%xlim([0 1]);
ylim([-4.0 6.0]);
axis square
grid on

cd(current_dir);
print(Plot_Task5, 'latex-meres2\fig\task5', '-depsc')

%% Task6-PI-valt-ref
cd('Exercise-5\Task6-PI-valt-ref');
Time      % t vector
Torque    % torque vector

```

```

Velocity    % velocity vector
Ref_signal  % ref vector

Plot_Task6 = figure('Position',[50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), velocity, (time./1000), ref, 'LineWidth', 1)
legend('Velocity','Ref signal');
xlabel('Time [s]','FontSize',12)
ylabel('Velocity [rad/s]','FontSize',12)
xlim([0 1]);
ylim([0 10]);
axis square
grid on

subplot(1,2,2)
plot( (time./1000), torque, 'b', 'LineWidth', 1)
xlabel('Time [s]','FontSize',12)
ylabel('Torque [mNm]','FontSize',12)
xlim([0 1]);
ylim([-6.0 6.0]);
axis square
grid on

cd(current_dir);
print(Plot_Task6,'latex-meres2\fig\task6','-depsc')

%% Task7-PI-terheles-ugras
cd('Exercise-5\Task7-PI-terheles-ugras');
Time        % t vector
Torque      % torque vector
Velocity    % velocity vector

Plot_Task7 = figure('Position',[50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
xlabel('Time [s]','FontSize',12)
ylabel('Velocity [rad/s]','FontSize',12)
xlim([0 1]);
%ylim([0 10]);
axis square
grid on

subplot(1,2,2)
plot( (time./1000), torque, 'b', 'LineWidth', 1)
xlabel('Time [s]','FontSize',12)
ylabel('Torque [mNm]','FontSize',12)
xlim([0 1]);
ylim([-4.0 6.0]);
axis square
grid on

cd(current_dir);
print(Plot_Task7,'latex-meres2\fig\task7','-depsc')

```

```

%% Task8-PI-ibatoleracio
cd('Exercise-5\Task8-PI-ibatoleracio');
Time      % t vector

Torque_a
torque_a = torque;
Torque_b
torque_b = torque;
Torque_c
torque_c = torque;
Torque_d
torque_d = torque;

Velocity_a
velocity_a = velocity;
Velocity_b
velocity_b = velocity;
Velocity_c
velocity_c = velocity;
Velocity_d
velocity_d = velocity;

Int_a
integral_a = integral;
Int_b
integral_b = integral;
Int_c
integral_c = integral;
Int_d
integral_d = integral;

Plot_Task8 = figure('Position',[50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), velocity_a, (time./1000), velocity_b, (time./1000), velocity_c, (time./1000), velocity_d);
legend('a eset','b eset','c eset','d eset');
xlabel('Time [s]','FontSize',12)
ylabel('Velocity [rad/s]','FontSize',12)
xlim([0 1]);
ylim([0 15]);
axis square
grid on

subplot(1,2,2)
plot( (time./1000), torque_a, (time./1000), torque_b, (time./1000), torque_c, (time./1000), torque_d);
legend('a eset','b eset','c eset','d eset');
xlabel('Time [s]','FontSize',12)
ylabel('Torque [mNm]','FontSize',12)
xlim([0 1]);
ylim([-6.0 6.0]);
axis square
grid on

```

```

Plot_Integral = figure;
plot( (time./1000), integral_a, (time./1000), integral_b, (time./1000), integral_c, (time./1000), integral_d);
legend('a eset', 'b eset', 'c eset', 'd eset');
%title('PCI 1784 számláló használata', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Integral', 'FontSize', 12)
xlim([0 1.0]);
%ylim([-1.5 1.5]);
axis square
grid on

cd(current_dir);
print(Plot_Task8, 'latex-meres2\fig\task8', '-depsc')
print(Plot_Integral, 'latex-meres2\fig\task8-Integral', '-depsc')

% Task9-PI-kesleltetett
cd('Exercise-5\Task9-PI-kesleltetett');
Time
Torque
Velocity
Int

Plot_Velocity = figure;
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Velocity [rad/s]', 'FontSize', 12)
xlim([0 1.0]);
%ylim([-1.5 1.5]);
axis square
grid on

Plot_Torque = figure;
plot( (time./1000), torque, 'b', 'LineWidth', 1)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Torque [mNm]', 'FontSize', 12)
xlim([0 1.0]);
ylim([-4.0 6.0]);
axis square
grid on

Plot_Integral = figure;
plot( (time./1000), integral, 'b', 'LineWidth', 1)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Integral', 'FontSize', 12)
xlim([0 1.0]);
%ylim([-1.5 1.5]);
axis square
grid on

cd(current_dir);
print(Plot_Velocity, 'latex-meres2\fig\task9-velocity', '-depsc')
print(Plot_Torque, 'latex-meres2\fig\task9-torque', '-depsc')
print(Plot_Integral, 'latex-meres2\fig\task9-integral', '-depsc')

```

```

%% Task10-P-PI
cd('Exercise-5\Task10-P-PI');
Time      % t vector
Pos_P1-I0
Velocity_P1-I0

Plot_a = figure('Position',[50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), position, 'b', 'LineWidth', 1)
title('P = 1, I = 0','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Position [rad]','FontSize',12)
xlim([0 5]);
ylim([0 10]);
axis square
grid on

subplot(1,2,2)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
title('P = 1, I = 0','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Velocity [rad/s]','FontSize',12)
xlim([0 5]);
ylim([-4.0 6.0]);
axis square
grid on

Pos_P05-I0
Velocity_P05-I0

Plot_b = figure('Position',[50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), position, 'b', 'LineWidth', 1)
title('P = 0.5, I = 0','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Position [rad]','FontSize',12)
xlim([0 5]);
ylim([0 10]);
axis square
grid on

subplot(1,2,2)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
title('P = 0.5, I = 0','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Velocity [rad/s]','FontSize',12)
xlim([0 5]);
ylim([-4.0 6.0]);
axis square
grid on

```

```

Pos_P05_I03
Velocity_P05_I03

Plot_c = figure('Position',[50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), position, 'b', 'LineWidth', 1)
title('P = 0.5, I = 0.3','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Position [rad]','FontSize',12)
xlim([0 5]);
%ylim([0 10]);
axis square
grid on

subplot(1,2,2)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
title('P = 0.5, I = 0.3','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Velocity [rad/s]','FontSize',12)
xlim([0 5]);
%ylim([-4.0 6.0]);
axis square
grid on

cd(current_dir);
print(Plot_a, 'latex-meres2\fig\task10_P1-I0', '-depsc')
print(Plot_b, 'latex-meres2\fig\task10_P05-I0', '-depsc')
print(Plot_c, 'latex-meres2\fig\task10_P05-I03', '-depsc')

%% Task11-Stick-slip
cd('Exercise-5\Task11-Stick-slip');
Time          % t vector
Pos_P01_I01
Torque_P01_I01
Velocity_P01_I01
Int_P01_I01

Plot_a = figure('Position',[0 0 1200 1024]);
subplot(2,2,1)
plot( (time./1000), position, 'b', 'LineWidth', 1)
title('P = 0.1, I = 0.1','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Position [rad]','FontSize',12)
xlim([0 10]);
%ylim([0 10]);
axis square
grid on

subplot(2,2,2)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
title('P = 0.1, I = 0.1','FontSize',10)
xlabel('Time [s]','FontSize',12)

```



```

ylabel('Velocity [rad/s]', 'FontSize', 12)
xlim([0 10]);
%ylim([-4.0 6.0]);
axis square
grid on

subplot(2,2,3)
plot( (time./1000), torque, 'b', 'LineWidth', 1)
title('P = 0.1, I = 0.1', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Torque [mNm]', 'FontSize', 12)
xlim([0 10]);
%ylim([-4.0 6.0]);
axis square
grid on

subplot(2,2,4)
plot( (time./1000), integral, 'b', 'LineWidth', 1)
title('P = 0.1, I = 0.1', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Integral', 'FontSize', 12)
xlim([0 10]);
%ylim([-4.0 6.0]);
axis square
grid on

Pos_P015_I02
Torque_P015_I02
Velocity_P015_I02
Int_P015_I02

Plot_b = figure('Position',[0 0 1200 1024]);
subplot(2,2,1)
plot( (time./1000), position, 'b', 'LineWidth', 1)
title('P = 0.15, I = 0.2', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Position [rad]', 'FontSize', 12)
xlim([0 10]);
%ylim([0 10]);
axis square
grid on

subplot(2,2,2)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
title('P = 0.15, I = 0.2', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Velocity [rad/s]', 'FontSize', 12)
xlim([0 10]);
%ylim([-4.0 6.0]);
axis square
grid on

subplot(2,2,3)
plot( (time./1000), torque, 'b', 'LineWidth', 1)

```

```

title('P = 0.15, I = 0.2', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Torque [mNm]', 'FontSize', 12)
xlim([0 10]);
%ylim([-4.0 6.0]);
axis square
grid on

subplot(2,2,4)
plot( (time./1000), integral, 'b', 'LineWidth', 1)
title('P = 0.15, I = 0.2', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Integral', 'FontSize', 12)
xlim([0 10]);
%ylim([-4.0 6.0]);
axis square
grid on

cd(current_dir);
print(Plot_a, 'latex-meres2\fig\task11_P01-I01', '-depsc')
print(Plot_b, 'latex-meres2\fig\task11_P015-I02', '-depsc')

% Task12-Poz-belseb-szab
cd('Exercise-5\Task12-Poz-belseb-szab');
Time
Pos_Ppos3_Pvel23_Ivel50
Torque_Ppos3_Pvel23_Ivel50
Velocity_Ppos3_Pvel23_Ivel50

Plot_positionA = figure;
plot( (time./1000), position, 'b', 'LineWidth', 1)
title('P- $\{pos\}$  = 3, P- $\{vel\}$  = 2.3, I- $\{vel\}$  = 50', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Position [rad]', 'FontSize', 12)
xlim([0 5]);
%ylim([-1.5 1.5]);
axis square
grid on

Plot_Task12A = figure('Position', [50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
title('P- $\{pos\}$  = 3, P- $\{vel\}$  = 2.3, I- $\{vel\}$  = 50', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Velocity [rad/s]', 'FontSize', 12)
xlim([0 5]);
ylim([-10 15]);
axis square
grid on

subplot(1,2,2)
plot( (time./1000), torque, 'b', 'LineWidth', 1)

```

```

title('P_{pos} = 3, P_{vel} = 2.3, I_{vel} = 50', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Torque [mNm]', 'FontSize', 12)
xlim([0 5]);
ylim([-6.0 6.0]);
axis square
grid on

Pos_Ppos15_Pvel23_Ivel50
Torque_Ppos15_Pvel23_Ivel50
Velocity_Ppos15_Pvel23_Ivel50

Plot_positionB = figure;
plot( (time./1000), position, 'b', 'LineWidth', 1)
title('P_{pos} = 1.5, P_{vel} = 2.3, I_{vel} = 50', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Position [rad]', 'FontSize', 12)
xlim([0 5]);
%ylim([-1.5 1.5]);
axis square
grid on

Plot_Task12B = figure('Position', [50 250 1024 450]);
subplot(1,2,1)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
title('P_{pos} = 1.5, P_{vel} = 2.3, I_{vel} = 50', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Velocity [rad/s]', 'FontSize', 12)
xlim([0 5]);
ylim([-10 15]);
axis square
grid on

subplot(1,2,2)
plot( (time./1000), torque, 'b', 'LineWidth', 1)
title('P_{pos} = 1.5, P_{vel} = 2.3, I_{vel} = 50', 'FontSize', 10)
xlabel('Time [s]', 'FontSize', 12)
ylabel('Torque [mNm]', 'FontSize', 12)
xlim([0 5]);
ylim([-6.0 6.0]);
axis square
grid on

cd(current_dir);
print(Plot_positionA, 'latex-meres2\fig\task12_posA', '-depsc')
print(Plot_Task12A, 'latex-meres2\fig\task12_vtA', '-depsc')
print(Plot_positionB, 'latex-meres2\fig\task12_posB', '-depsc')
print(Plot_Task12B, 'latex-meres2\fig\task12_vtB', '-depsc')

%% Task13-Csuszomod
cd('Exercise-5\Task13-Csuszomod\Csuszomod-1');
Time

```

```

Pos
Velocity
Torque
Sigma

Plot_a = figure('Position',[0 0 1200 1024]);
subplot(2,2,1)
plot( (time./1000), position, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Position [rad]','FontSize',12)
xlim([0 10]);
ylim([0 10]);
axis square
grid on

subplot(2,2,2)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Velocity [rad/s]','FontSize',12)
xlim([0 10]);
ylim([-4.0 6.0]);
axis square
grid on

subplot(2,2,3)
plot( (time./1000), torque, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Torque [mNm]','FontSize',12)
xlim([0 10]);
ylim([-4.0 6.0]);
axis square
grid on

subplot(2,2,4)
plot( (time./1000), sigma, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Sigma','FontSize',12)
xlim([0 10]);
ylim([-4.0 6.0]);
axis square
grid on

Plot_ErrorA = figure;
plot( 5-position, -velocity, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó','FontSize',10)
xlabel('Position error [rad]','FontSize',12);
ylabel('Velocity error [rad/s]','FontSize',12);
xlim([0 5]);
ylim([-4.0 6.0]);
axis square
grid on

```

```

cd(current_dir);
print(Plot_a, 'latex-meres2\fig\task13_csusz1', '-depsc')
print(Plot_ErrorA, 'latex-meres2\fig\task13_csusz1_Error', '-depsc')

cd('Exercise-5\Task13-Csuszomod\Csuszomod-2-kotyog');
Time
Pos
Velocity
Torque
Sigma

Plot_Kotyog = figure('Position',[0 0 1200 1024]);
subplot(2,2,1)
plot( (time./1000), position, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó, kotyogás','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Position [rad]','FontSize',12)
xlim([0 12]);
ylim([0 10]);
axis square
grid on

subplot(2,2,2)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó, kotyogás','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Velocity [rad/s]','FontSize',12)
xlim([0 12]);
ylim([-4.0 6.0]);
axis square
grid on

subplot(2,2,3)
plot( (time./1000), torque, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó, kotyogás','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Torque [mNm]','FontSize',12)
xlim([0 12]);
ylim([-4.0 6.0]);
axis square
grid on

subplot(2,2,4)
plot( (time./1000), sigma, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó, kotyogás','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Sigma','FontSize',12)
xlim([0 12]);
ylim([-4.0 6.0]);
axis square
grid on

```

```

Plot_Koty_Error = figure;
plot( 5-position, -velocity, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó, kotyogás','FontSize',10)
xlabel('Position error [rad]','FontSize',12);
ylabel('Velocity error [rad/s]','FontSize',12);
xlim([0 5]);
%ylim([-4.0 6.0]);
axis square
grid on

cd(current_dir);
print(Plot_Kotyog,'latex-meres2\fig\task13_csusz2_koty','-depsc')
print(Plot_Koty_Error,'latex-meres2\fig\task13_csusz2_koty_Error','-depsc')

cd('Exercise-5\Task13-Csuszomod\Csuszomod-2-nincs-kotyog');
Time
Pos
Velocity
Torque
Sigma

Plot_NKotyog = figure('Position',[0 0 1200 1024]);
subplot(2,2,1)
plot( (time./1000), position, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó, nincs kotyogás','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Position [rad]','FontSize',12)
xlim([0 12]);
%ylim([0 10]);
axis square
grid on

subplot(2,2,2)
plot( (time./1000), velocity, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó, nincs kotyogás','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Velocity [rad/s]','FontSize',12)
xlim([0 12]);
%ylim([-4.0 6.0]);
axis square
grid on

subplot(2,2,3)
plot( (time./1000), torque, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó, nincs kotyogás','FontSize',10)
xlabel('Time [s]','FontSize',12)
ylabel('Torque [mNm]','FontSize',12)
xlim([0 12]);
%ylim([-4.0 6.0]);
axis square
grid on

subplot(2,2,4)

```

```

plot( (time./1000), sigma, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó, nincs kotyogás', 'FontSize',10)
xlabel('Time [s]', 'FontSize',12)
ylabel('Sigma', 'FontSize',12)
xlim([0 12]);
%ylim([-4.0 6.0]);
axis square
grid on

Plot_NKoty_Error = figure;
plot( 5-position, -velocity, 'b', 'LineWidth', 1)
title('Csúszómód szabályozó, nincs kotyogás', 'FontSize',10)
xlabel('Position error [rad]', 'FontSize',12);
ylabel('Velocity error [rad/s]', 'FontSize',12);
xlim([0 5]);
%ylim([-4.0 6.0]);
axis square
grid on

cd(current_dir);
print(Plot_NKotyog, 'latex-meres2\fig\task13_csusz2_Nkoty', '-depsc')
print(Plot_NKoty_Error, 'latex-meres2\fig\task13_csusz2_Nkoty_Error', '-depsc')

```