

Obsah

Řízení výkonu NPA R04 - uživatelská příručka	1
Obsah	1
1. Jak řízení funguje (přehled)	1
Rychlá orientace v menu	2
2. Nastavení rozpadového místa	2
2.1 Základní údaje	2
2.2 Způsob regulace zařízení (lo_regDev)	3
2.3 Mapování povelů Px na procenta (lo_rgP0...lo_rgP4)	3
2.4 Priorita vypínání (lo_regPwrPriority)	3
2.5 Řízení externím PLC přes Modbus (lo_regExtPLC)	4
3. Pověly a omezení	4
3.0 SoP - Celkový instalovaný výkon	4
3.1 Pověly distribučního dispečinku (DŘ)	4
3.2 Omezení přetoku do sítě	4
3.3 Spotové ceny	5
4. Nastavení regulátorů	5
4.1 Význam konstant	5
4.2 Vzorec regulátoru	6
5. Příklady s grafy	6
5.1 Přesné doregulování vs. okamžitý povel	6
5.2 Měníče s velkým dopravním zpožděním	7
5.3 Co se stane při kmitání (špatné parametry)	8
5.4 Asymetrie i+ / i- - rychlé stažení přetoku, pomalý náběh	8
6. Postup ladění (cookbook)	9
Zdrojové soubory	9

Řízení výkonu NPA R04 - uživatelská příručka

Platí pro verzi NPA R04: 2.13.beta20 · Vytvořeno: 2026-06-17

Tento dokument vysvětluje, jak jednotka NPA R04 řídí **činný výkon** výroby: jak se nastavují **rozpadová místa**, jak se zadávají **pověly a omezení** a jak se ladí **regulátory**. Na konci jsou praktické **příklady s grafy** - včetně přesného doregulování na procenta, řízení měničů s velkým dopravním zpožděním a ukázky, kdy regulátor začne kmitat.

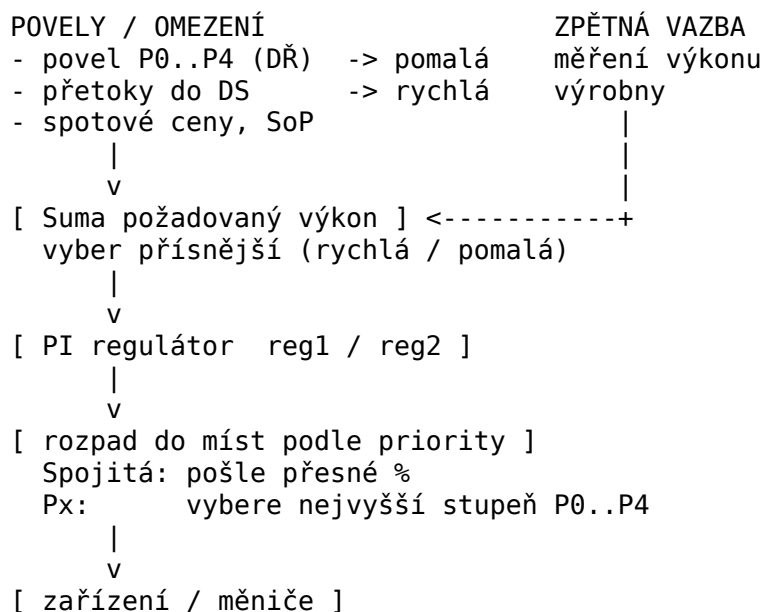
Tato příručka popisuje regulaci **činného výkonu** (modul NRegulation2). **Řízení BESS, cos φ a jalového výkonu sem nepatří** - mají samostatné dokumenty (BESS viz bess.md, cos φ a jalový výkon samostatně).

Obsah

1. Jak řízení funguje (přehled)
2. Nastavení rozpadového místa
3. Pověly a omezení
4. Nastavení regulátorů
5. Příklady s grafy
6. Postup ladění (cookbook)

1. Jak řízení funguje (přehled)

Regulační smyčka běží **každých 500 ms** (NRegulation2::Loop). V každém kroku:



- **Požadovaný výkon** se počítá ze dvou větví:
 - **pomalý** - z povelu distribuce P0..P4 (= PowerLimit v %),
 - **rychlý** - z omezení **přetoku** do sítě (povolený přetok, SoP rezervovaný výkon, spotové ceny).
- Regulátor použije tu **přísnější** větev: je-li rychlý požadavek ≤ pomalý, řídí **rychlý regulátor přetoků** (reg2), jinak **pomalý regulátor povelů** (reg1).
- Výsledné procento se rozdělí do **rozpadových míst** podle priority a podle toho, zda místo umí **spojité** řízení nebo jen **stupňové povelů Px**.

Rychlá orientace v menu

Veškeré nastavení se dělá ve webovém rozhraní jednotky (horní menu). Kde co najdete:

Co nastavujete	Cesta v menu
Vlastnosti místa, způsob řízení, povel Px (lo_*)	Rozpadová místa → klik na místo → <i>editace</i>
SoP rezervovaný výkon/příkon, úrovně P0-P4 (fs_sopRsrvPwr, fs_rgP0..4)	Nastavení výrobní
Omezení přetoku do sítě (fs_flwRest...)	Nastavení výrobní
Konstanty regulátorů činného výkonu (fs_reg1/2_k/i+/i-), spotové ceny (fs_TarifPacLimit...)	Nastavení výrobní → dole <i>Nastavení placených funkcí</i>

Editovat hodnoty placených funkcí (včetně konstant regulátorů) může jen účet s vyšším oprávněním; běžný uživatel je vidí jen ke čtení.

2. Nastavení rozpadového místa

Menu: Rozpadová místa → klikněte na konkrétní místo → tlačítko *editace*.

2.1 Základní údaje

Pole	Klíč	Význam
Typ	lo_typeDs	regulaci činného výkonu řeší typy FVE, MVE, KGJ, BPS, VTE, DA, HE (BESS, TUV, AKU, Wallbox mají vlastní řízení)
Instalovaný výkon	lo_insPower	špičkový instalovaný výkon [Wp]
Skutečný činný výkon	lo_Power	jmenovitý činný výkon měničů [W]

Maximální výkon místa pro regulaci = $\min(\text{lo_Power}, \text{lo_insPower})$ (když je lo_insPower zadán), jinak lo_Power. Procenta se vždy počítají z tohoto maxima.

2.2 Způsob regulace zařízení (lo_regDev)

Toto je klíčové rozhodnutí podle toho, **co zařízení v daném místě umí**:

Volba	Klíč	Kdy použít
Spojité regulace	Spojita	Zařízení umí přijmout libovolné % (měniče FVE, KGJ s plynulým řízením...). Regulátor pošle přesné procento .
Regulace povelů Px	Px	Zařízení neumí plynulé řízení – jen pevné stupně (relé, přepínání odboček, starší měniče reagující jen na povelové úrovně). Regulátor vybere nejvyšší stupeň , jehož výkon nepřekročí požadované %.
Bez regulace	none	Místo se nereguluje. Vypne se (0 %) jen pokud je požadováno < 100 %.

Px = „jen pro zařízení, která neumožňují plynulé řízení“. Místo plynulého procenta se používá 5 pevných stupňů P0...P4 (viz dále). Regulátor nikdy nenastaví procento „mezi“ stupni – sníží se na nejbližší **nižší nebo rovný** stupeň, aby nedošlo k překročení limitu.

2.3 Mapování povelů Px na procenta (lo_rgP0...lo_rgP4)

Distribuce posílá povelové úrovně P0...P4. Pro každou úroveň se v místě nastaví, **na kolik procent** výkonu se má místo omezit:

Úroveň	Klíč	Typický význam
P0	lo_rgP0	např. 100 % (plný výkon)
P1	lo_rgP1	např. 60 %
P2	lo_rgP2	např. 30 %
P3	lo_rgP3	např. 10 %
P4	lo_rgP4	např. 0 % (stop)

Globální výchozí hodnoty z továrního nastavení se zobrazují vedle (fs_rgP0...fs_rgP4). Tlačítkem **Nastavit** u každého stupně lze povel ručně otestovat (setlocationlevelp). Pole **Manuální nastavení výkonu** (lo_rgGlobal, povel „Px“) nastaví místo ručně na zadané % (setlocationpower).

2.4 Priorita vypínání (lo_regPwrPriority)

Když je nutné omezit výkon, místa se obsluhují podle priority (0 = nejvyšší přednost zůstat v provozu):

- **Zapíná se od nejnižšího čísla, vypíná se od nejvyššího.**
- Místo s **nižším číslem** priority dostane výkon dřív a **vypíná se jako poslední**.
- Místo s **vyšším číslem** se **omezuje/vypíná jako první**.
- Místa se stejnou prioritou a stejným lo_regDev tvoří **skupinu** regulovanou společně; spojitá místa se v rámci stejné priority řadí za stupňová (Px).

2.5 Řízení externím PLC přes Modbus (lo_regExtPLC)

Volba	Klíč	Význam
Neaktivní	none	bez vlivu
Řízení výkonu v %	regP_prc	nadřazené PLC zapisuje DS_power_value [%]
Řízení výkonu v kW	regP_kw	nadřazené PLC zapisuje Power_export_limit

Z lokálního požadavku a hodnoty od PLC se vždy bere **přísnější** (nižší) hodnota.

3. Povelý a omezení

Menu: Nastavení výroby (spotové ceny jsou dole v *Nastavení placených funkcí*).

Zdroje požadovaného výkonu:

3.0 SoP - Celkový instalovaný výkon

Pole	Klíč	Význam
SoP - Celkový instalovaný výkon	fs_sopTotalPwr	celkový instalovaný výkon výroby dle Smlouvy o připojení [W] – jen ke čtení , dopočítává se automaticky

Jak se hodnota počítá nyní: pole je v menu zablokované (disabled) a nezadává se ručně. Hodnota se dopočítává v C++ a neukládá jako **součet „Instalovaný výkon“ (lo_insPower) všech rozpadových míst** (s fallbackem na lo_Power, pokud je lo_insPower = 0).

Změna od verze 2.13.beta18 (2026-06-11). Dříve šlo o **ručně editovatelnou** položku: uživatel zadával celkový instalovaný výkon do pole a hodnota se ukládala do configu pod klíčem fs_sopTotalPwr. Regulace ji ale nikdy nečetla – pro výpočty vždy používala GetInstalovanyVykony() (součet lo_insPower), takže ručně zadaná hodnota mohla být s realitou nekonzistentní. Od beta18 je pole jen ke čtení a ukládaný klíč fs_sopTotalPwr se navíc **při načtení configu z configu maže** (hodnota se vždy dopočítá), aby v něm nezůstávala zastaralá ručně zadaná data.

3.1 Povelý distribučního dispečinku (DŘ)

Povel chodí z dispečinku po sběrnici/Modbusu, v menu se nezadává. Mapování úrovní na % se nastavuje u jednotlivých míst (viz 2.3).

- Povel P0...P4 → PowerLimit v % → **pomalý** požadovaný výkon = PowerLimit% × instalovaný výkon.
- Mapování úrovně na % řeší každé místo přes lo_rgP0...4 (viz 2.3).

3.2 Omezení přetoku do sítě

Menu: Nastavení výroby → *Nastavení hodnot omezení činného výkonu výroby a SoP.*

Pole	Klíč	Význam
Systémové řízení přetoků	fs_flwRestsw	zapnutí regulace přetoku (true/false)

Pole	Klíč	Význam
Povolený přetok	fs_flwRest	maximální dovolený přetok do DS [W]
Celkový rezervovaný výkon (SoP)	fs_sopRsrvPwr	smluvní maximum přetoku – nejvyšší priorita, aplikuje se vždy (i při vypnutém fs_flwRestsw)
Měření	fs_mRs	fs_measPhase = regulace po fázích (najde fázi s největším přetokem), jinak podle sumy P

Přetok = záporná spotřeba na hlavním elektroměru (TML). Regulátor drží přetok na/pod povolenou hodnotou → **rychlá** větev.

3.3 Spotové ceny

Menu: Nastavení výroby → *Nastavení placených funkcí* (SPOT ceny).

Pole	Klíč	Význam
Tarifní limit	fs_TarifPacLimit	zapnutí řízení podle spotových cen
Cena nákupu (distribuce)	fs_TarifPacLimitRate	[Kč/MWh]
Cena prodeje (obchodník)	fs_TarifRedemptionPacLimitRate	[Kč/MWh]
Regulovat na	fs_TarifPacLimitValue	cílový přetok při nevýhodné ceně

Při nevýhodné výkupní ceně se přetok omezí (nebo se při záporných cenách výkon stáhne na 0 – „vyplatí se i mařit“). Bere se vždy **přísnější** z nastaveného přetoku a přetoku podle cen.

Jalový výkon / cos ϕ a BESS mají vlastní regulátory (reg3, reg4) a jsou popsány v samostatných dokumentech (BESS viz bess.md). V této příručce je neřešíme.

4. Nastavení regulátorů

Menu: Nastavení výroby → dole *Nastavení placených funkcí* → tabulka *Nastavení regulátoru*. Konstanty mají u sebe nápovědu (najetím myši na pole). Editace jen pro účet s vyšším oprávněním.

Činný výkon řídí **dva** PI regulátory (další dva – reg3 pro cos ϕ a reg4 pro BESS – mají samostatné dokumenty):

#	Název v UI	Klíče	Co reguluje
1	Regulátor pro povel Px z DŘ	fs_reg1_k, fs_reg1_i+, fs_reg1_i-	pomalá regulace podle povelů distribuce
2	Regulátor pro přetoky do DS	fs_reg2_k, fs_reg2_i+, fs_reg2_i-	rychlá regulace přetoku

4.1 Význam konstant

Každý regulátor má tři konstanty:

- **k – proporcionální konstanta (P)**. Působí **okamžitě** úměrně regulační odchylce.
 - $k = 0 \rightarrow$ **P složka vyřazena**; odchylku koriguje **jen integrační složka** (nad rámec stále přítomné dopředné vazby na žádanou hodnotu, viz 4.2). Výchozí stav.

- Malé k (řádově 0,001-0,01) už má silný vliv (odchylka je ve wattech, tj. velká čísla).
- Příliš velké $k \rightarrow$ překmity až nestabilita.
- **i+ - integrační konstanta pro stoupání** (odchylka > 0 , tj. měření je **pod** žádanou hodnotou $\rightarrow$ výkon **roste**).
- **i- - integrační konstanta pro klesání** (odchylka < 0 , tj. měření je **nad** žádanou hodnotou $\rightarrow$ výkon **klesá**).
- Integrační konstanta je **časová konstanta**: čím **menší** číslo, tím **rychlejší** přibližování (jednotky = velmi rychlé, ale hrozí překmit; desítky = pomalé a stabilní).
- $i = 0 \rightarrow$ **integrační složka vyřazena a integrátor vynulován**.

Asymetrie $i+ \neq i-$ umožňuje např. **rychle stáhnout** přetok a **pomalů najíždět** zpět. Výchozí regulátor přetoků má $i- = 5$ (rychlý pokles) a $i+ = 20$ (pomalý nárůst).

4.2 Vzorec regulátoru

Regulátory běží v každém kroku (0,5 s). Regulace činného výkonu (reg1, reg2) má tvar s **dopřednou vazbou** (výstup startuje na žádané hodnotě a PI ji jen koriguje):

```
odchylka      = žádaná - měřená
integrátor    += odchylka / i+      (když odchylka > 0)
integrátor    += odchylka / i-      (když odchylka < 0)
```

výstup = žádaná + (žádaná / y_max) * (k * odchylka + integrátor)

y_max = maximální výkon na fázi (instalovaný výkon / 3).

Důležité vlastnosti:

- **Integrátor odstraní trvalou odchylku** – akumuluje se, dokud měření $\neq$ žádaná hodnota. Proto i při $k = 0$ regulátor **dojede přesně** na cílovou hodnotu (přesné doregulování).
- **Při $i+ = i- = 0$** je integrace vyřazena, takže výstup = žádaná – na výstup jde **přímo procento z povelu** a odchylka se **vůbec nedoreguluje** (okamžitý povel, viz 5.1 B).
- Při $\text{žádaná} = 0$ (vypnutá regulace) a při nulovém měření se **integrátor resetuje** (jinak by „nafoukl“ a po zapnutí způsobil překmit).
- **Anti-windup**: při saturaci výstupu (na y_min/y_max) se integrátor po fázích zmrazí.

5. Příklady s grafy

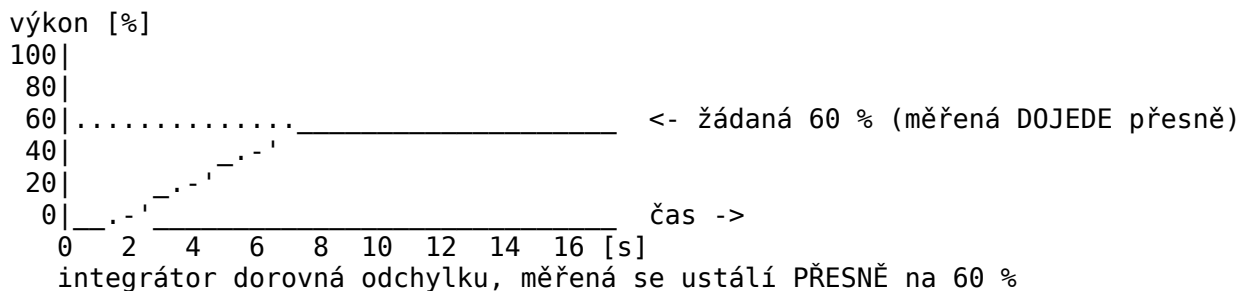
Grafy jsou ilustrační (perioda kroku 0,5 s). Žádaná hodnota = **60 %**, místo se rozbíhá z 0.

5.1 Přesné doregulování vs. okamžitý povel

Podle nastavení integračních konstant se regulátor chová **dvěma zcela odlišnými způsoby**. Rozdíl je v tom, **zda se vůbec řeší odchylka** mezi žádanou a skutečně naměřenou hodnotou.

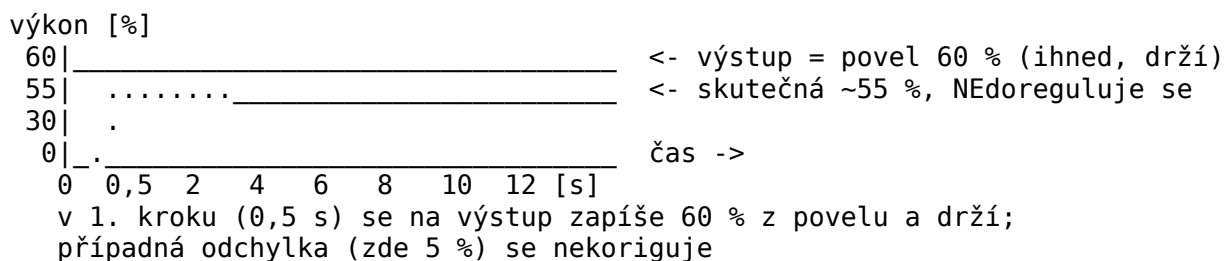
A) Přesné doregulování (zpětnovazební) Regulátor **sleduje odchylku** a integrační složkou ji postupně vynuluje – skutečně naměřený výkon **přesně dojde** na žádané procento (dorovná i ztráty, derating, nepřesnost měničů). Je o něco **pomalejší**, protože musí doregulovat.

```
fs_reg1_k  = 0      (P vyřazeno)
fs_reg1_i+ = 20      (doregulování nahoru)
fs_reg1_i- = 20      (doregulování dolů)
```



B) Okamžitý povel (rychlý, bez doregulování odchylky) Vyřadí i integrační složku (vše na 0). Pak $k \cdot \text{odchylka} + \text{integrátor} = 0$ a podle vzorce platí **výstup = žádaná**. Na výstup se v **prvním kroku ihned přepíše procento z povelu** a drží – **odchylka se vůbec neřeší** (žádná zpětná vazba). Nejrychlejší možná reakce, ale skutečně naměřená hodnota se **nedoreguluje** (zůstane, jaká vyjde).

fs_reg1_k = 0 (P vyřazeno)
 fs_reg1_i+ = 0 (integrace vyřazena)
 fs_reg1_i- = 0 (integrace vyřazena)



Shrnutí: $i+/i- > 0 \rightarrow$ **přesné doregulování** (měřená dojde na cíl, pomalejší); $i+ = i- = 0 \rightarrow$ **okamžitý povel** (na výstup se ihned zapíše % z povelu, odchylka se neřeší). Proportionální k nech v obou případech na 0.

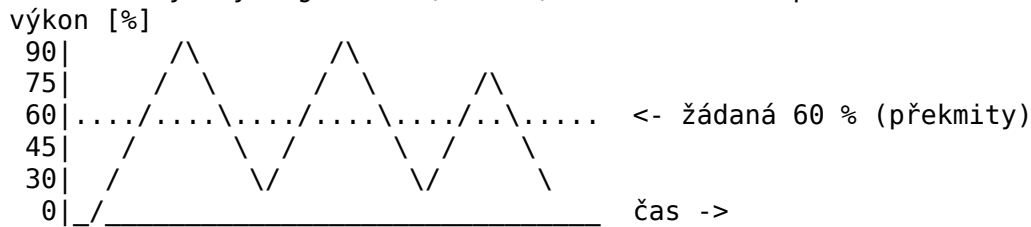
5.2 Měníče s velkým dopravním zpožděním

Problém: měnič na změnu žádané hodnoty zareaguje až za několik sekund (dopravní zpoždění / mrtvá doba). Agresivní regulátor „nevidí“ efekt svého zásahu, přidá ještě víc – a **přestřelí**.

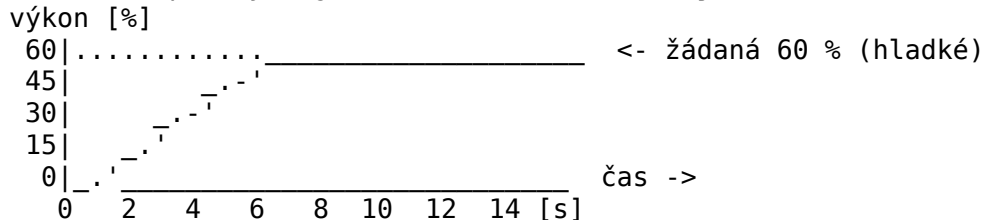
Řešení: regulátor **zpomalit** – integrační konstanta musí být **výrazně větší** než dopravní zpoždění. P složku držet na nule:

fs_reg2_k = 0 (žádné okamžité skoky)
 fs_reg2_i+ = 40 (pomalé, opatrné najíždění)
 fs_reg2_i- = 30

ŠPATNĚ - rychlý regulátor ($i+ = 3$) na měniči se zpožděním 3 s:



SPRÁVNĚ - pomalý regulátor ($i+ = 40$) na stejném měniči:

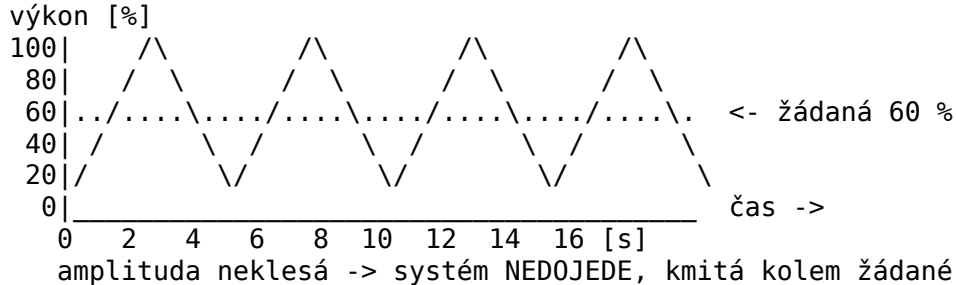


Pravidlo: čím větší dopravní zpoždění měniče, tím **větší** $i+/i-$ (pomalejší regulátor).
Orientačně volte integrační konstantu několikanásobek mrtvé doby.

5.3 Co se stane při kmitání (špatné parametry)

Když je k příliš velké nebo i příliš malé vzhledem k dynamice soustavy (a zpoždění), regulátor se dostane do **trvalého kmitání** až nestability:

$fs_reg2_k = 0,05$ <- příliš velké
 $fs_reg2_i+ = 2$ <- příliš rychlé

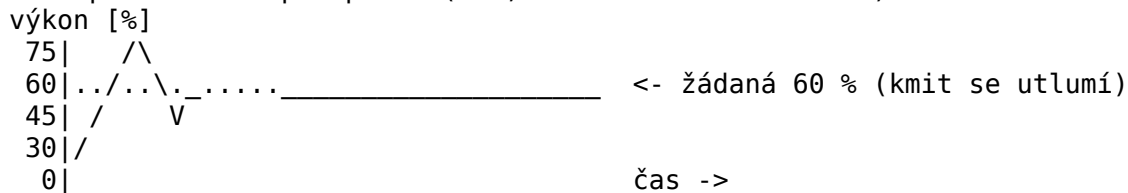


Projevy kmitání: výkon i přetok „pendlují“, měniče přepínají nahoru/dolů, na elektroměru oscilace.

Náprava:

1. Sniž k (ideálně na 0).
2. Zvětš $i+/i-$ (zpomalení integrace).
3. Pokud kmitání přetrvává, zkontroluj dopravní zpoždění měničů (viz 5.2).

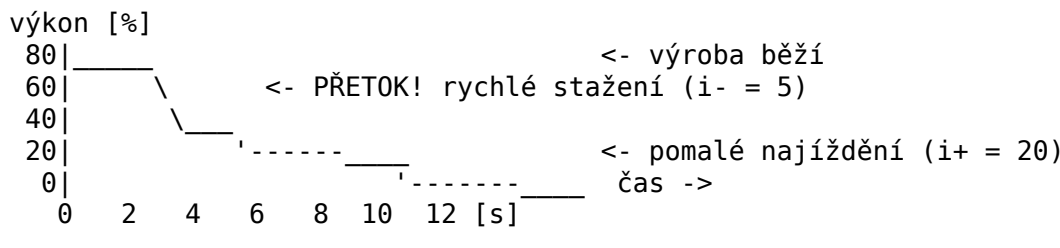
Postupné tlumení po opravě ($k=0$, $i+$ zvětšeno z 2 na 25):



5.4 Asymetrie $i+ / i-$ - rychlé stažení přetoku, pomalý náběh

Pro regulaci přetoku se hodí **rychle stáhnout** výkon (bezpečnost – nepřekročit limit) a **pomalů najíždět** zpět (klid sítě):

```
fs_reg2_k = 0
fs_reg2_i+ = 20 (pomalý nárůst)
fs_reg2_i- = 5 (rychlý pokles)
```



6. Postup ladění (cookbook)

1. **Začni od továrního nastavení** - $k = 0$, integrace v desítkách (reg1 $i_{\pm} = 20$, reg2 $i_{+} = 20$, $i_{-} = 5$). Tento stav je stabilní a dojde přesně.
2. **Příliš pomalá reakce?** Postupně **zmenšuj** i_{+}/i_{-} (např. $20 \rightarrow 10$). Po každé změně sleduj překmit.
3. **Objeví se překmit/kmitání?** Vrať i zpět na vyšší hodnotu a/nebo nech $k = 0$. k zvyšuj jen výjimečně a po malých krocích ($0,001 \rightarrow 0,005 \rightarrow 0,01$).
4. **Měníče reagují se zpožděním?** Drž $k = 0$ a i nastav na **několikanásobek** mrtvé doby (viz 5.2).
5. **Přetoky musí být přísné?** Nastav i_{-} (pokles) výrazně menší než i_{+} (nárůst) - viz 5.4.
6. **Zařízení neumí plynulé řízení?** Přepni místo na $lo_regDev = Px$ a vyplň $lo_rgP0...4$.

Situace	k	i_{+}	i_{-}
Přesné doregulování (měřená dojde na cíl)	0	20	20
Okamžitý povel (ihned zapíše %, neřeší odchylku)	0	0	0
Rychlejší doregulování (svižné měniče)	0	8-12	8-12
Velké dopravní zpoždění měničů	0	30-60	30-60
Přísné přetoky (rychle dolů, pomalu nahoru)	0	20	5
△ Nestabilní (NEPOUŽÍVAT)	$\geq 0,05$	≤ 2	≤ 2

Zlaté pravidlo: pro přesnost a stabilitu nech $k = 0$ a lad' **jen** integračními konstantami i_{+}/i_{-} . Proporcionální složku přidávej jen, když potřebuješ rychlejší okamžitou reakci a víš, že soustava nemá velké zpoždění.

Zdrojové soubory

Část	Soubor / symbol
Hlavní smyčka regulace	nregulation2.cpp → NRegulation2::Loop()
PI regulátor (činný výkon)	npicontroller.cpp → updateParam()
Rozpad do míst, priorit, Px	nregulation2.cpp → GetLocationPrioritySort(), GroupSetPower()
Povely distribuce / přetoky / SoP / spotové ceny	nregulation2.cpp → RegulacePodleDistribuce(), RegulacePodlePretoku(), RegulacePodleSpotCen()
Výchozí konstanty regulátorů	nregulation2.h → NR_CONST[]
UI - nastavení místa	html/v2/location_edit.html
UI - nastavení regulátorů	html/v2/factory_settings_edit_paid_features.html