

7 Обробка вихідних даних контурів регулювання (сімейство *Output Processing*)

7.1 Блок управління реверсивним двигуном (*SERVO*)

Функціональний блок *SERVO* призначений для реалізації управляючих дій з регуляторів *PI_B/PIDFF* або інших блоків з використанням виконавчих механізмів типу реверсивного двигуна (серводвигун, наприклад *MEO*). Для управління серводвигунами використовуються два виходи - *RAISE* ("більше") та *LOWER* ("менше"), на яких сигнал формується в залежності від значення входу *IN* (або *INPD*) та налаштування блоку (рис.7.1). У таблиці 7.1 показані параметри блоку *SERVO*.

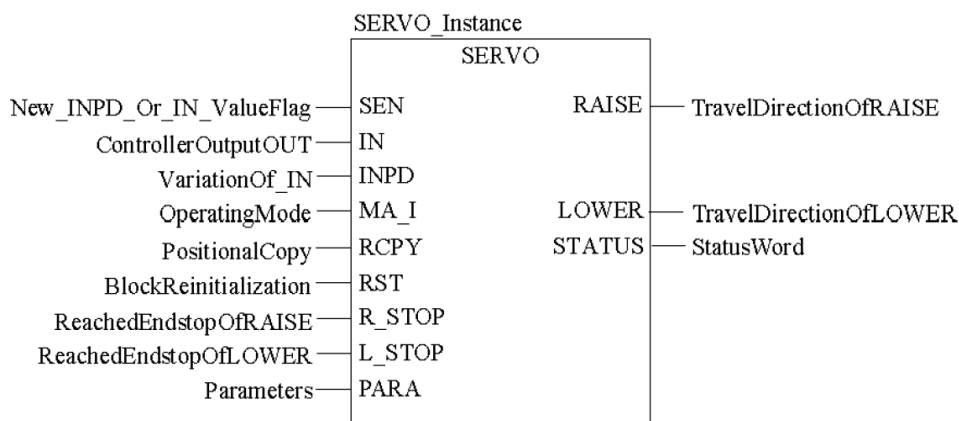


Рис.7.1. Виклик функціонального блоку *SERVO*.

Таблиця 7.1 Параметри блоку *SERVO*

Вхідні параметри		
<i>SEN</i>	<i>BOOL</i>	1 : є нові значення на входах <i>INPD</i> або <i>IN</i> 0 : немає нових значень на входах <i>INPD</i> або <i>IN</i>
<i>IN</i>	<i>REAL</i>	значення з абсолютного виходу блоку управління <i>OUT</i> (0 до 100%)
<i>INPD</i>	<i>REAL</i>	значення з інкрементального виходу блоку управління <i>OUTD</i> (-100% до 100%)
<i>MA_I</i>	<i>BOOL</i>	значення операційного режиму блоку регулятора до якого підключений <i>SERVO</i> (вихід <i>MA_O</i>) 1 : Автоматичний режим 0 : інший режим (ручний або слідкування)
<i>RCPY</i>	<i>REAL</i>	Зворотній зв'язок по позиції ВМ (0 до100%)
<i>RST</i>	<i>BOOL</i>	1 : Ініціалізація функціонального блоку (відключення виходів та обнуління внутрішнього статусу блоку)
<i>R_STOP</i>	<i>BOOL</i>	кінцева позиція "відкритий" (досягнутий поріг по <i>RAISE</i>)
<i>L_STOP</i>	<i>BOOL</i>	кінцева позиція "закритий" (досягнутий поріг по <i>LOWER</i>)
<i>PARA</i>	<i>Para_SERVO</i>	Параметри (див. таб 6.24)
Вихідні параметри		
<i>RAISE</i>	<i>BOOL</i>	Вихід "більше" ("відкрити")
<i>LOWER</i>	<i>BOOL</i>	Вихід "менше" ("закрити")
<i>STATUS</i>	<i>WORD</i>	Слово статусу

Таблиця 7.2 Опис структурного типу *Para_SERVO*

<i>en_rcpy</i>	<i>BOOL</i>	1 : Використовується зворотній зв'язок по позиції ВМ (включити <i>RCPY</i>)
<i>rcpy_rev</i>	<i>BOOL</i>	1 : Інвертувати <i>RCPY</i> (0 – це 100%, 100 – це 0%) 0 : Не інвертувати <i>RCPY</i>
<i>t_motor</i>	<i>TIME</i>	час повного відкриття регулюючого органу
<i>t_mini</i>	<i>TIME</i>	мінімальна довжина імпульсу

Функціональний блок може використовуватись як з сигналом зворотного зв'язку по положенню виконавчого механізму (*RCPY*) так і без нього.

При використанні сигналу зворотного зв'язку (*en_rcpy = TRUE*), в якості управляючого входу *SERVO* треба використовувати вхід *IN*, який повинен бути зв'язаний з абсолютним виходом *OUT* регулятора *PI_B/PIDFF*. Для кожного нового значення виходу *OUT*, який формує регулятор, функціональний блок *SERVO* генерує дискретний вихід *RAISE* або *LOWER* з довжиною імпульсу пропорційною різниці *IN-RCPY*. При необхідності, вхід *RCPY* можна інвертувати (*rcpy_rev=1*), тобто коли показник положення регулюючого органу показує не процент відкриття, а процент закриття. Приклад діаграми роботи *SERVO* зі зворотним зв'язком по положенню ВМ показаний на рис. 7.2.

Якщо зворотний сигнал по положенню ВМ не використовується (*en_rcpy = FALSE*) то в якості управляючого входу *SERVO*, треба використовувати вхід *INPD*, який повинен бути зв'язаний з виходом *OUTD* регулятора *PI_B/PIDFF*. Для кожного нового значення виходу *OUTD* регулятора, блок *SERVO* генерує дискретний вихід *RAISE* або *LOWER* з довжиною імпульсу пропорційною *INPD*. Для правильного функціонування операційного режиму, вхід *MA_I* блоку *SERVO* повинен бути з'єднаний з виходом *MA_O* блоку регулятора.

Для формування імпульсу з тривалістю, пропорційною величині управляючого сигналу, необхідно вказати час повного відкриття регулюючого органу (*t_motor*). Тобто, наприклад, при формуванні на вході *INPD* значення 100%, блок *SERVO* сформує імпульс на виході *RAISE* рівним значенню *t_motor*. А при *INPD* рівним -10%, *SERVO* сформує імпульс на виході *LOWER* рівним значенню *t_motor/10*.

Для зменшення навантаження на двигун, тобто збільшення його терміну служби, задається мінімальний час імпульсу (*t_mini*). Якщо тривалість розрахованого імпульсу менша ніж *t_mini*, імпульс формуватися не буде, однак він буде врахований на наступних циклах.

Для точності регулювання, блок *SERVO* повинен викликатися з кожним циклом Задачі. Однак весь контур регулювання може оброблятися з меншою частотою, наприклад при використанні *SAMPLETM*. Для того щоб вказати блоку *SERVO* що регулятор *PI_B/PIDFF* оброблений і сформовані нові значення (особливо це стосується виходу *OUTD* регулятора), і його вихід *OUT/OUTD* а отже і вхід *SERVO IN/INPD* отримав нові значення, на вхід *SEN* подається *TRUE*. Таким чином, якщо в контурі для задавання періодичності обробки використовується блок *SAMPLETM*, то його вихід підключається паралельно до входів *EN* блоків *PI_B/PIDFF* і до входу *SEN* блоку *SERVO*.

У ручному режимі (*MA_I=FALSE*) блок *SERVO* обробляє входи *IN/INPD* в кожному циклі незалежно від значення входу *SEN*, що треба враховувати при

написанні програми. Тобто при ручній зміні виходу *OUT* регулятора, він автоматично розрахує *OUTD*, який протримається на виході аж до наступного виклику регулятора. За цей час, зв'язаний з цим виходом вхід *INPD* блоку *SERVO*, буде оброблений декілька (*n*) раз, що затягне імпульс в *n* раз довше, ніж потрібно. Для подолання цієї проблеми, в ручному режимі після кожного виклику регулятора необхідно обнулювати змінну, прив'язану до *OUTD* та *INPD*.

При використанні сигналу зворотного зв'язку (*en_rcpy = TRUE*) у ручному режимі блок *SERVO* буде видавати команди "більше" та "менше" на виконавчий механізм доти, поки входи *RCPY* та *IN* не будуть рівними. Це може негативно сказатися на процесі а також на роботі приводу та регулюючого органу. При досягненні крайнього положення регулюючим органом, тобто коли *R_STOP=TRUE* або *L_STOP=TRUE*, відповідний вихід виставляється в логічний нуль (*RAISE=FALSE* або *LOWER=FALSE*) незалежно від значення *IN* або *INPD*.

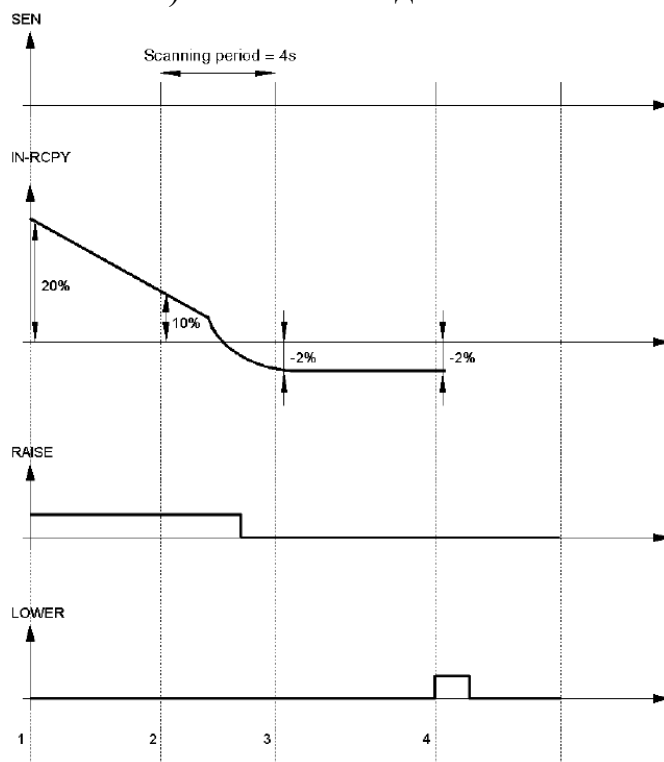


Рисунок 7.2– Приклад діаграми роботи блоку *SERVO* зі зворотним зв'язком по положенню ВМ.

Розглянемо роботу *SERVO* в режимі використання зворотного зв'язку по положенню, тобто при *en_rcpy=TRUE* (рис.7.2). У цьому прикладі використовуються такі настройки: $t_{motor}=t\#25s$, $t_{mini}=t\#1s$, періодичність виклику основного регулятора (періодичність активації *SEN*) - 4s. Контур працює в автоматичному режимі (*MA_I=TRUE*). На діаграмі цифрами мітками позначені наступні ситуації.

1. При початковому скануванні контуру значення (*IN-RCPY*)= $+20\%$: на виході *RAISE* генерується імпульс тривалістю 5s ($=20\%$ від 25s);

2. При наступному скануванні контур значення (*IN-RCPY*)= $+10\%$: на виході *RAISE=TRUE* залишається ще протягом 2.5s ($=10\%$ від 25s); одна секунда, яка залишилась від попереднього разу вже не враховується;

3. На 3-му циклі сканування контуру (*IN-RCPY*)= -2% : це відповідає імпульсу 0.5s на виході *LOWER*, однак оскільки це менше ніж t_{mini} , то вихід

$LOWER=FALSE$; тривалість 0.5s для $LOWER$ залишається збереженим до наступного циклу перерахунку;

На 4-му циклі сканування контуру ($IN-RCPY$)= -2%: це відповідає імпульсу 0.5s на виході $LOWER$, враховуючи попередній імпульс 0.5s, загальна тривалість імпульсу 1s, тобто на виході $LOWER=TRUE$ протягом однієї секунди.

Розглянемо роботу $SERVO$ без використання зворотного зв'язку по положенню, тобто при $en_rcpy=FALSE$ (рис.7.3). У цьому прикладі використовуються такі настройки: $t_motor=t\#25s$, $t_mini=t\#1s$, контур працює в ручному режимі ($MA_I=FALSE$). На діаграмі цифрами мітками позначені наступні ситуації.

1. При $INPD=+20\%$: на виході $RAISE$ генерується імпульс тривалістю 5s (=20% від 25s);

2. При появі значення $INPD=+2\%$: це відповідає імпульсу 0.5s на виході $RAISE$, однак оскільки це менше ніж t_mini , то вихід $RAISE=FALSE$; тривалість 0.5s для $RAISE$ залишається збереженим до наступного перерахунку;

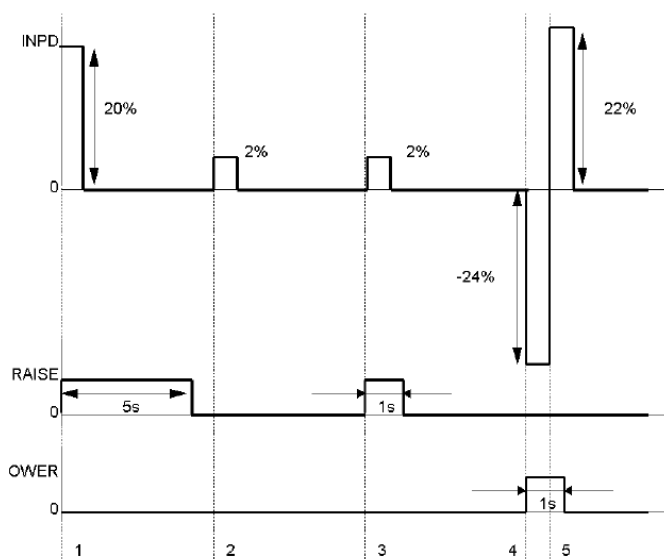


Рисунок 7.3— Приклад діаграми роботи блоку $SERVO$ без зворотного зв'язку по положенню ВМ.

3. При появі значення $INPD=+2\%$: це відповідає імпульсу 0.5s на виході $RAISE$, враховуючи попередній імпульс 0.5s, загальна тривалість імпульсу 1s, тобто на виході $RAISE=TRUE$ протягом однієї секунди;

4. При появі значення $INPD= -24\%$: це відповідає імпульсу 6s на виході $LOWER$;

5. Перед закінченням наступної секунди наступна модифікація $INPD=+22\%$: загальна модифікація = $-24\%+22\%=-2\%$, це менше ніж мінімальний імпульс (4%), тому вихід $LOWER=FALSE$.

7.2 Управління виконавчими механізмами з використанням широтно- імпульсного перетворення PWM

Функціональний блок PWM призначений для перетворення числового значення на вході блоку IN в 2 дискретні сигнали OUT_POS та OUT_NEG з певною шириною та частотою імпульсу (рис.7.4).

Ініціалізація блоку обов'язково повинна проводитись при першому циклі ПЛК!

Весь час ділиться на періоди з тривалістю t_period (рис.7.5). У залежності від значення входу IN , на кожному періоді формується сигнал OUT_POS (при $IN > 0$) або OUT_MIN (при $OUT < 0$) з тривалістю T_on . Залежність T_on від значення IN описується формулами (7.1)-(7.2), та графічно показана на рис.7.6.

$$T_on(OUT_POS) = t_period \cdot \frac{IN}{in_max} \quad \text{при } 0 \leq IN \leq in_max$$

$$T_on(OUT_NEG) = t_period \cdot \frac{|IN|}{in_max} \quad \text{при } 0 \leq -IN \leq in_max$$

(7.1), (7.2)

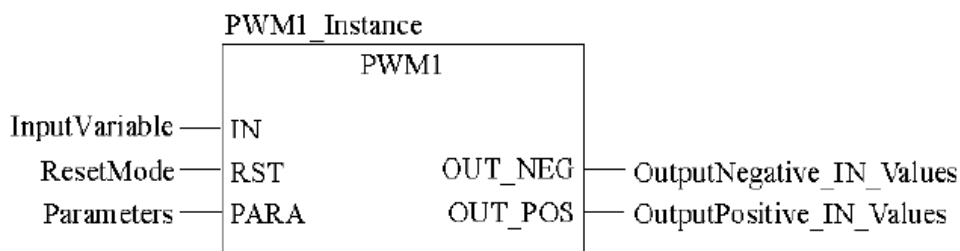


Рисунок 7.4– Приклад виклику блоку PWM1.

У таблиці 7.3 та 7.4 наведені параметри блоку PWM1.

Таблиця 7.3 Параметри блоку PWM1

Вхідні параметри		
IN	<i>REAL</i>	Вхідна змінна
RST	<i>BOOL</i>	1 = Скинути блок
$PARA$	<i>Para_PWM1</i>	Параметри
Вихідні параметри		
OUT_NEG	<i>BOOL</i>	Вихід для від'ємного значення IN
OUT_POS	<i>BOOL</i>	Вихід для додатного значення IN

Таблиця 7.4 Опис структурного типу $Para_PWM1$

t_period	<i>TIME</i>	Довжина (тривалість) періоду
t_min	<i>TIME</i>	Мінімальний час імпульсу
in_max	<i>REAL</i>	обмеження по максимуму (по модулю) значення IN

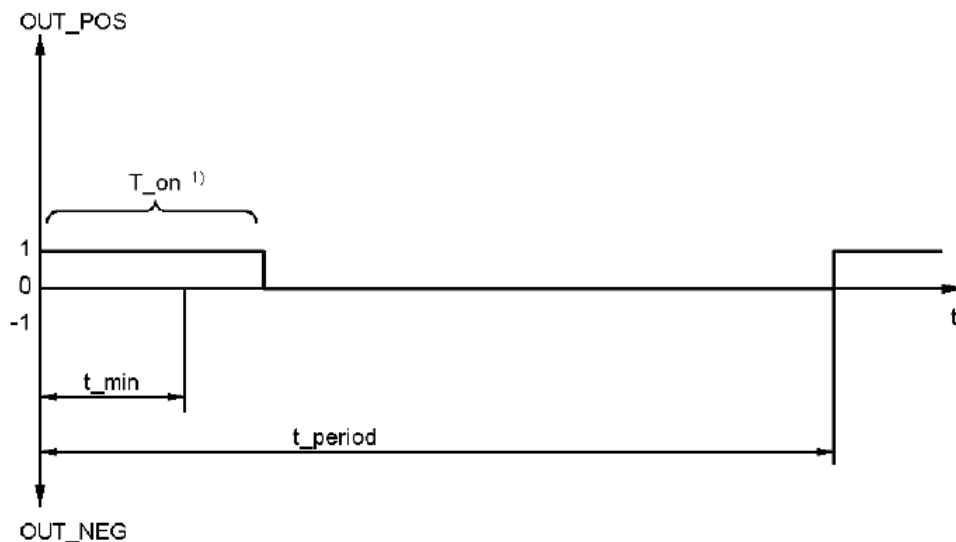


Рисунок 7.5– Діаграма зміни виходів блоку *PWM1*.

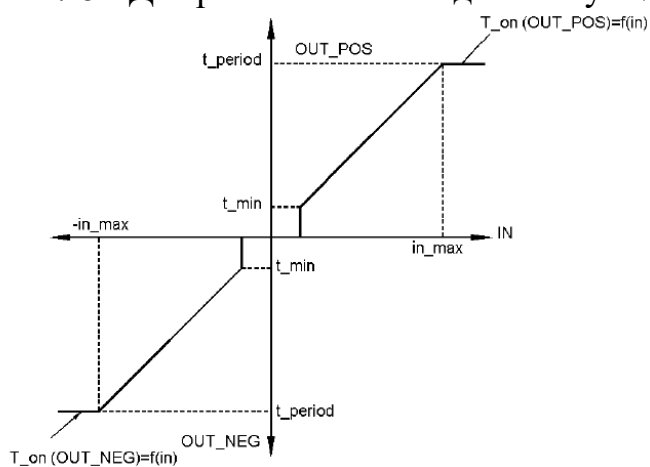


Рисунок 7.6– Статична характеристика $t_period(IN)$ для *PWM1*.

Параметр t_min вказує мінімальну тривалість імпульсу. Тобто якщо розрахований час T_on за (7.1) та (7.2) буде менше часу t_min , то імпульс на даному періоді формуватися не буде.

При команді $RST=1$ всі виходи блоку скидаються в *FALSE*, а блок починає відраховувати час з моменту, коли $RST=0$.

Якщо блок *PWM1* обробляє вихід регуляторів *PI_B/PIDFF*, то рекомендується t_period виставляти рівним часу виклику блоку регулятора. Це гарантує, що кожний новий сигнал на ВМ з регулятора буде оброблений блоком *PWM1*. Час періоду виклику самого *PWM1* рекомендується виставляти як мінімум в 10 раз менше ніж час періоду t_period .

Блок *PWM1* може використовуватися для управління двигуном, швидкість якого керується шириною імпульсу *OUT_POS* в один бік, і *OUT_NEG* в інший бік. Іншим прикладом виконавчого механізму є *ТЕН* (трубчатий електронагрівач), однак при цьому буде використовуватись тільки вихід *OUT_POS*.

Приклад діаграми роботи *PWM1* з настройками $t_period=t\#4s$, $t_min=t\#0.5s$, $in_max=10$ показаний на рис.7.7.

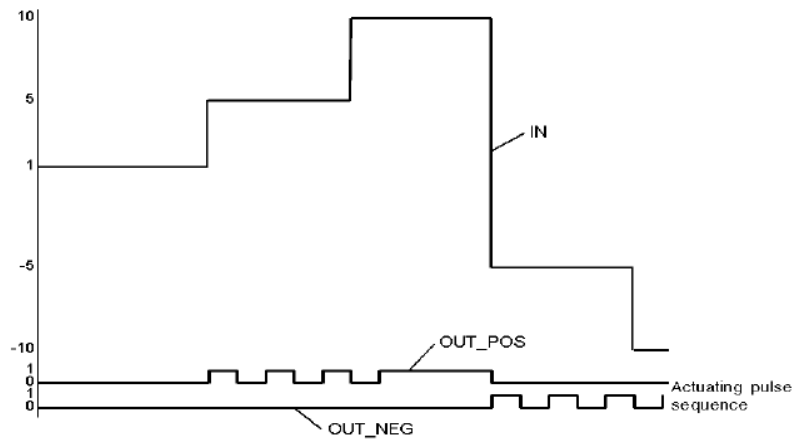


Рисунок 7.7– Приклад діаграми роботи PWM1.

4.3.3. Блок ручного управління *MS*

Функціональні блоки регуляторів, в тому числі *PI_B/PIDFF*, мають вбудовані засоби переключення їх в ручний режим, з забезпеченням безперебійності. Нижче наведені випадки, для яких може знадобитися використання блоку ручного управління *MS*:

- для управління аналоговими виходами, які управляються НЕ через алгоритми зі зворотним зв'язком (не за відхиленням);
- для управління аналоговими виходами, які в ручному і в автоматичному режимах формуються окремими блоками (наприклад при каскадному регулюванні, переключенні між алгоритмами);
- для управління серводвигунами в ручному режимі без використання регуляторів *PI_B/PIDFF*;

На рис.7.8 показана функціональна схема блоку *MS*, на рис.7.9 – приклад виклику блоку в FBD.

Вихід *OUT* в автоматичному режимі формується як сума входів *IN* та *outbias*, якщо зміщення активоване (*use_bias=TRUE*). У ручному режимі вихід міняється безпосередньо. В обох режимах абсолютне значення *OUT* обмежується по мінімуму і максимуму, а інкрементальне *OUTD* – не обмежується.

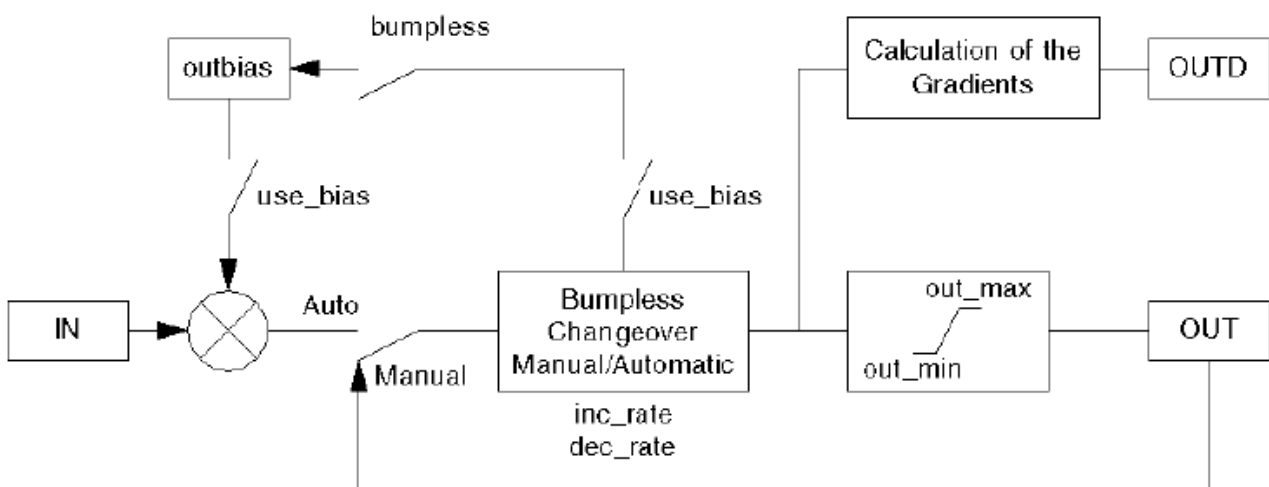


Рис.7.8.Функціональна схема блоку *MS*.

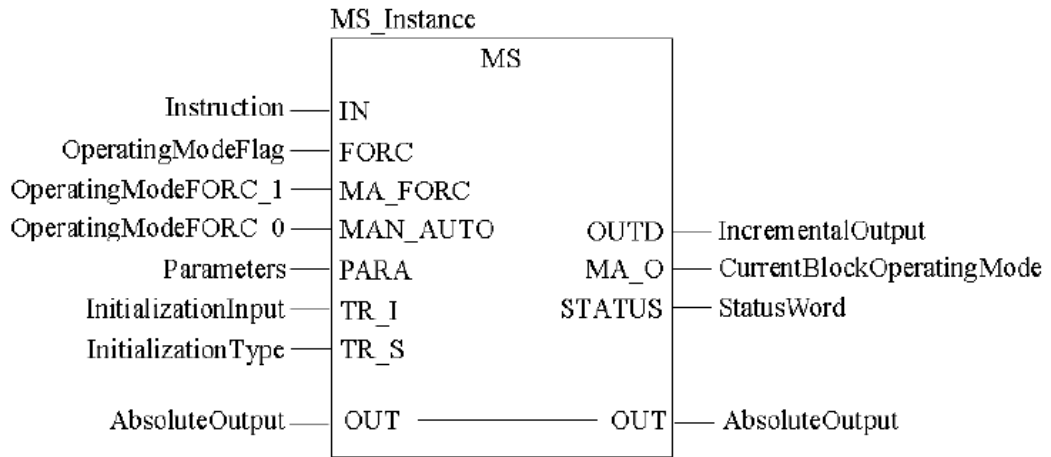


Рисунок 7.9—Приклад виклику блоку *MS*

Таблиця 7.5 Параметри блоку *MS*

Вхідні параметри		
<i>IN</i>	<i>REAL</i>	Значення змінної в автоматичному режимі
<i>FORC</i>	<i>BOOL</i>	1: Режим ручний/автомат визначається входом <i>MA_FORC</i> 0: Режим ручний/автомат визначається входом <i>MAN_AUTO</i>
<i>MA_FORC</i>	<i>BOOL</i>	Режим ручний/автомат для <i>FORC = 1</i> 1: Автоматичний режим 0: Ручний режим
<i>MAN_AUTO</i>	<i>BOOL</i>	Режим ручний/автомат для <i>FORC = 0</i> 1: Автоматичний режим 0: Ручний режим
<i>PARA</i>	<i>Para_MS</i>	Параметр
<i>TR_I</i>	<i>REAL</i>	Вхід ініціалізації
<i>TR_S</i>	<i>BOOL</i>	Команда на включення ініціалізації (1: Включити вхід ініціалізації)
Вхідні/вихідні параметри		
<i>OUT</i>	<i>REAL</i>	абсолютне значення виходу
Вихідні параметри		
<i>OUTD</i>	<i>REAL</i>	інкрементальне значення виходу: різниця між вихідною величиною в поточному і попередньому циклах перерахунку <i>OUTD</i>
<i>MA_O</i>	<i>BOOL</i>	Плинний режим роботи регулятора (0: Ручний, 1: Автоматичний)
<i>STATUS</i>	<i>WORD</i>	Слово статусу

Таблиця 7.6 Опис структурного типу *Para_MS*

<i>out_min</i>	<i>REAL</i>	нижня межа для виходу регулятора
<i>out_max</i>	<i>REAL</i>	верхня межа для виходу регулятора
<i>inc_rate</i>	<i>REAL</i>	Швидкість наростання значення сигналу при переключенні руч./авт. (одиниць на секунду)
<i>dec_rate</i>	<i>REAL</i>	Швидкість спадання значення сигналу при переключенні руч./авт. (одиниць на секунду)
<i>outbias</i>	<i>REAL</i>	значення зміщення
<i>use_bias</i>	<i>BOOL</i>	1: Активувати зміщення
<i>bumpless</i>	<i>BOOL</i>	1: активувати безударність переходу руч/авт

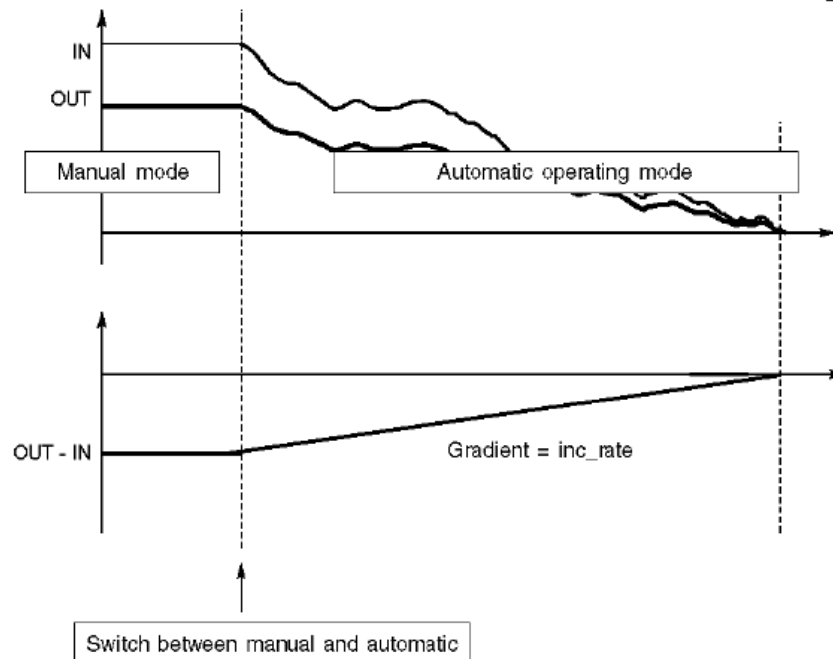


Рисунок 7.10–Діаграма роботи безударного переходу руч./авт. в *MS*.

Для реалізації безударності переходу з ручного в автоматичний режим (*bumpless=TRUE*) використовуються лінійні переходи з обмеженням по швидкості, які настроюються параметрами *inc_rate* та *dec_rate*. Таким чином, в момент переходу в автоматичний режим, різниця між *IN* та *OUT* буде зменшуватися відповідно до заданої швидкості (рис.7.10).

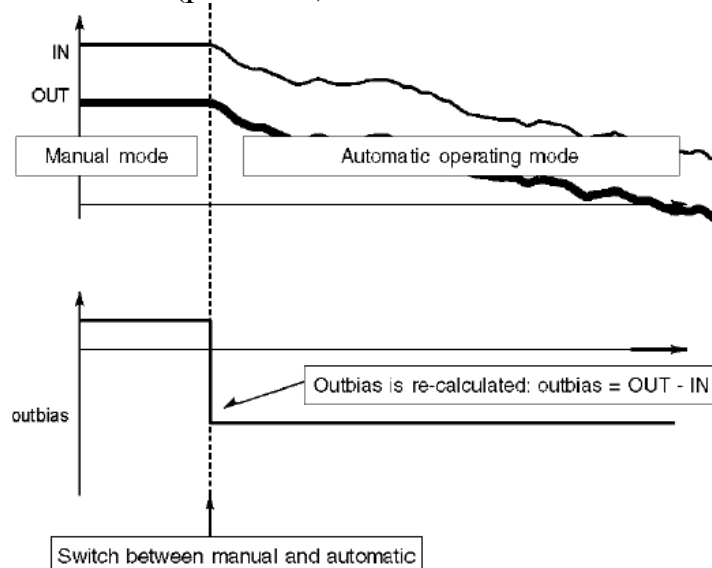


Рисунок 7.11. Діаграма роботи безударного переходу руч./авт.

зі зміщенням в *MS*.

Якщо активований параметр *outbias* (зміщення) тобто *use_bias=TRUE*, то безударність досягається за рахунок використання нового перерахованого значення зміщення (рис.7.11).

Безударний перехід є сенс активувати тоді, коли вхід *IN* підключений до регуляторів або інших блоків, в яких не реалізований безударний перехід.