

КОНТРОЛЕР ЗА МИКРО DC СЕРВОДВИГАТЕЛ

Тодор Илиев Неделчев

Технически Университет - Габрово, ул. Хаджи Димитър № 4, Габрово, България
todnedh@abv.bg

MICRO DC SERVO MOTOR CONTROLLER

Todor Iliev Nedelchev

Technical University of Gabrovo, Hadji Dimitar str. 4, Gabrovo, Bulgaria
todnedh@abv.bg

Abstract

Micro DC servomotors have very low rotor inertia, high torque and small time constants. For this reason, they find application in computer equipment such as tape drives, printers, disk drives and word processors, robotic systems and small CNC machines that require precision speed control and precise positioning. This study presents a study on the development of a servomotor controller SMC with AVR 32-bit RISC processors ESP32 and in a simplified version with a low-cost 8-bit microcontroller AT90S2313. Simulation results made with MathLab Simulink are presented.

Keywords: servomotor controller SMC, PID tuning, DC servomotor.

ВЪВЕДЕНИЕ

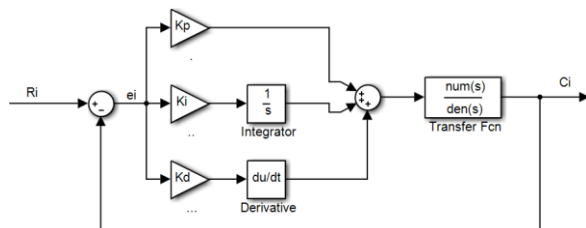
DC серво двигателят е самостоятелна електромеханична система, която завърта изходния вал до определена ъглова позиция в отговор на електрически сигнал, проектиран е за прецизен контрол на позицията, скоростта и ускорението. Състои се от DC двигател, свързан с управляваща верига и датчик за обратна връзка, като например енкодер. Конфигурацията позволява непрекъснато регулиране на състоянието на двигателя, затова намира голямо приложение в устройства, които изискват прецизност и динамична производителност.

Управляваща верига интерпретира входните сигнали (обикновено показващи желаната скорост или позиция), като съответно регулира тока или напрежението, подавани към двигателя. Системата за обратна връзка постоянно следи

състоянието на двигателя, предавайки данни обратно към управляващата верига, за да приведе действителното състояние на двигателя в съответствие с желаните параметри, осигурявайки по-висока точност и прецизно управление.

В промишлеността се използват много видове DC серводвигатели, защото имат много малка инерция на ротора, което води до висок динамичен въртящ момент, малки времеви константи и относително малки номинални мощности. Затова те се използват в компютърно оборудване, роботиката и CNC машини (Computer Numerical Control), като лентови устройства, принтери, дискови устройства, текстови процесори и манипулатори[1], които изискват точност на управление на скоростта и точно позициониране. При серводвигателите позицията и скоростта на двигателя се контролират

от сигнали, изпращани от контролера с обратна връзка [2]. През последните години в индустриалните процеси се използват Пропорционално-интегрално-диференциални (PID) контролери (фиг. 1), тъй като те постигат минимизиране на грешките чрез промяна на входа за управление на процеса [3].



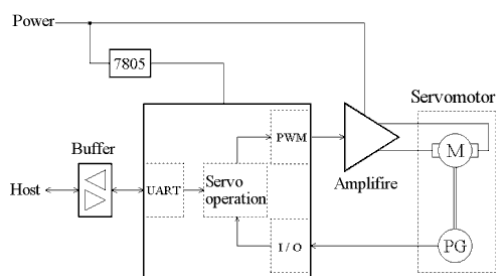
Фиг. 1. Блокова схема на PID контролера

$$Ri(T) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

Където Ri е входът на PID контролера, който има три коефициенти на усилване (K_p , K_i и K_d), а грешката е (ei).

Едно от предимствата на PID контролера е качеството на неговата работа като основен контролер с непрекъсната обратна връзка.

Блоковата схема на контролер за DC серводвигател с непрекъсната обратна връзка е показана на фиг. 2 [5].



Фиг. 2. Контролер за DC серводвигател-блокова схема

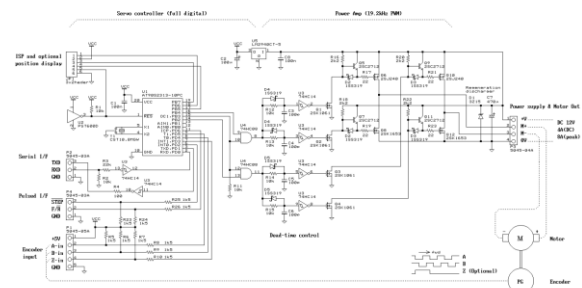
ИЗЛОЖЕНИЕ

Голямото приложение на DC серводвигатели (servomotor control system SMC) създава предпоставка за разработване система за управление със затворен контур. Тя изисква серво операции в реално време, като например управление

на позицията, управление на скоростта и управление на въртящия момент. Системата за управление е подходяща за внедряване във всякакви вградени AVR 32-битови RISC процесори (Например ESP32) като междинен софтуер[4]. В опростен проект тези операции могат да се обработват само с евтин 8-битов микроконтролер (AT90S2313).

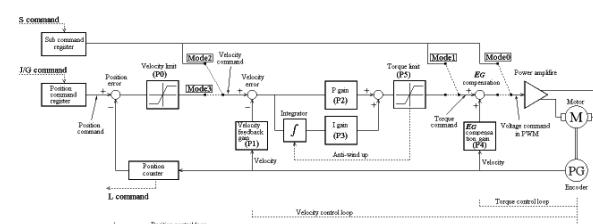
Към контролера се включва усилвател на мощност, високоефективен драйвер с който може да управлява двигател с мощност до 50 W при захранващо напрежение 12 V.

На фиг. 3 е показана принципната схема на контролер за DC серводвигател. JP1 е ISP конектор за програмиране на AVR и може да се използва и за свързване на LED дисплей. P2 е хост интерфейс, който се управлява от всеки хост контролер и може да се свърже директно към серийния порт на компютър. P4 е импулсно командван интерфейс, подобен на повечето серво контролери.



Фиг. 3. Принципна схема на SMC

Конфигурация с множествена обратна връзка, която е най-популярният и основен серво режим в момента е показана на фиг. 4 [2]. Работата на серво системата с множествена обратна връзка се нарича „каскадно управление“.



Фиг. 4. Диаграма на работата на SMC

При каскадно управление, един или няколко управляващи сигнала, реакцията на които е по-бърза от тази на основния контур, се групират в основния контур за управление като второстепенен контур за управление, по този начин се увеличава общата производителност на сервомотора [5].

Серво операциите се обработват от софтуерно имплементиран цифров серво оператор. За да се направи дискретна апроксимация на серво работата, е необходим период на актуализиране, който е много по-бърз от механичния отговор. Обикновено се избира честота на актуализиране от 1 kHz до няколко kHz. SMC обработва серво операциите с честота на актуализиране 1 kHz, като всяка операция се обработва в рамките на 92 μ с, така че AVR може лесно да се използва за серво обработка.

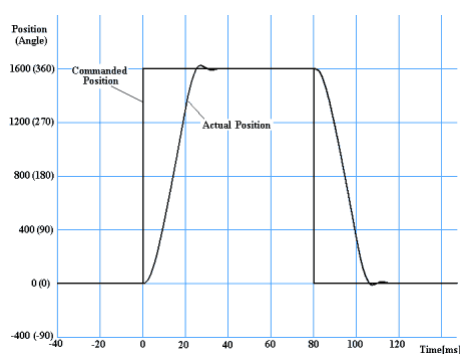
SMC може да работи в четири различни серво режима:

- 0 (режим с управление на напрежението) задвижва двигателя с постоянно напрежение;

- 1 (режим с управление на въртящия момент) задвижва двигателя с постоянен въртящ момент, независимо от скоростта на въртене;

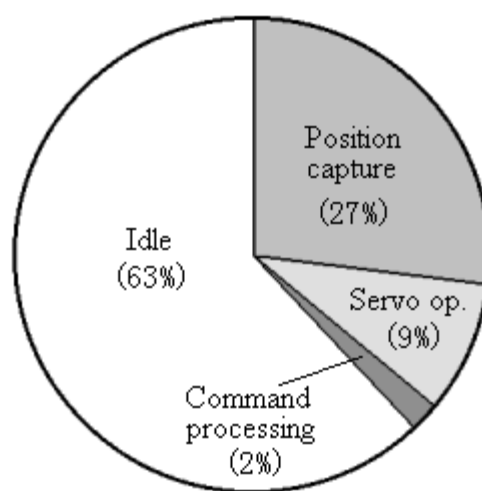
- 2 (режим с управление на скоростта) задвижва двигателя с постоянна скорост, независимо от въртящия момент на натоварване;

- 3 (режим на позициониране) задвижва двигателя до зададената позиция и я задържа там. Това се нарича "следящо серво" и е типично за системи за управление на движението.



Фиг. 5. Стъпка на реакция на SMC

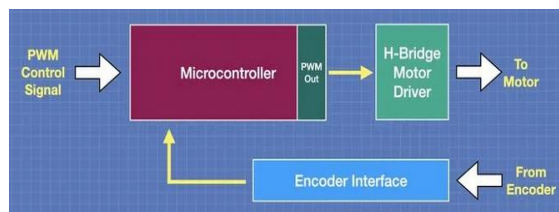
Фигура 5 показва измерената стъпкова характеристика при работа на серводвигателя с проследяване (режим 3) с параметри $V_S = 14.5V$, $K_F = 4.6$, $K_P = 1.5$, $K_I = 0.13$. Това е траекторията на движението на 360-градусово скачащо движение. От графиката се вижда, че времето за настройка е по-малко от 40 ms и се появява леко превишаване. За да се използва ефективно SMC за управление на движението, параметрите на сервото трябва да бъдат настроени за действителните стойности на натоварването.



Фиг. 6. Коефициент на натоварване на процесора

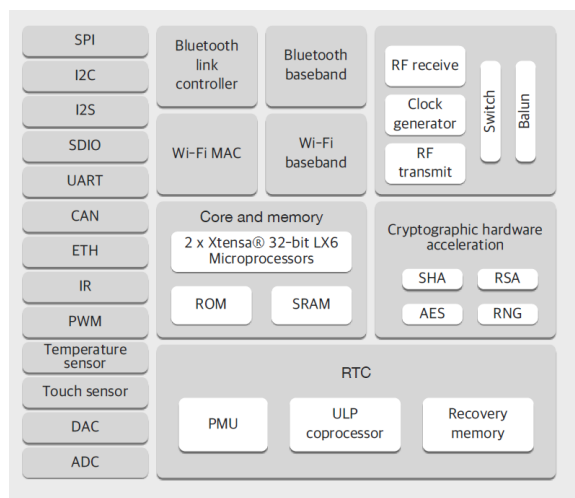
Фигура 6 показва коефициента на натоварване на процесора за всеки процес. Тъй като SMC улавя входните данни от енкодера само със софтуер, този процес заема най-много време от ефективното време за обработка. Софтуерното улавяне на данни от енкодера може да се използва най-добре на две оси. Серво обработката изглежда правилна, въпреки че обработва много операции. Следователно, над 60 процента от мощността на процесора е в режим на покой, AVR има достатъчен капацитет за серво управление. Той ще може да се използва за серво управление до няколко оси, ако се добави външен буферен брояч за улавяне на данни от енкодера.

За управление на малки DC серводвигатели за неиндустриални цели е подходящ контролера ESP 32 с драйвери PCA9685/TB6612FNG [8]. Блоквата схема илюстрира работата на цифров SMC е показана на фиг. 7.



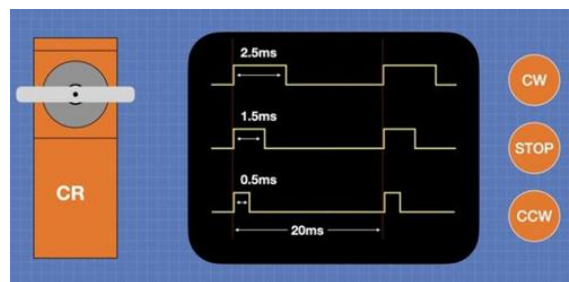
Фиг. 7. Блокова схема на цифров SMC

ESP32-WROOM-32 е мощен, универсален Wi-Fi+BT+BLE MCU модул, насочен към широк спектър от приложения, В основата на този модул е чипът ESP32-D0WDQ6*[6]. Вграденият чип е проектиран да бъде мащабируем и адаптивен. Има две процесорни ядра, които могат да се управляват индивидуално, а тактовата честота на процесора е регулируема от 80 MHz до 240 MHz. Може да се изключи процесора и да използва нискоенергийния копроцесор, за да се наблюдават периферните устройства за промени или преминаване на прагове. Блоквата схема на контролера е показана на фиг. 8.



Фиг. 8. Блокова схема на EPS 32

Малките DC серво се управляват чрез широчинно-импулсна модулация (ШИМ). Обикновено се използва по-втарящ се импулсен сигнал, работещ с 50 Hz (импулс на всеки 20 милисекунди) [8]. Ширината на високата част на импулса в рамките на всеки период от 20 милисекунди определя целевия ъгъл на сервото. Ширина на импулса между 0,5 и 1 μ s съответства на единия край на диапазона на сервото, докато ширина на импулса между 2 и 2,5 μ s съответства на другия край. Всяка ширина на импулса, попадаща между тези стойности, ще премести сервото в междинна позиция. Импулс с ширина от 1,5 μ s центрира вала и ще спре двигателя. Увеличаването на ширината на импулса ще завърти двигателя по посока на часовниковата стрелка (CW Clockwise). Колкото по-широк е импулсът, толкова по-бързо ще се върти, докато достигне максимална скорост с импулс от 2 до 2,5 μ s. Импулс с ширина под 1,5 μ s завърта двигателя обратно на часовниковата стрелка (CCW- Counter clockwise). С намаляването на ширината на импулса той се върти по-бързо, докато достигне максимална скорост при минималната ширина на импулса на двигателя (фиг. 9).



Фиг. 9. ШИМ за управление на DC серводвигател

За да се задвижи вала на двигателя напред-назад е направен тестов код на C++ за SMC (фиг. 10).

Включва се библиотеката ESP32Servo и се създава обект „myservo“, който да представя сервомотора. Разпределяме и серво пина GPIO 8 на ESP32 [8]. В настройката се разпределят и четирите

PWM таймера на ESP32. След това се задава PWM периода (честотата) 50 Hz и минималната и максималната ширина на PWM импулса. Използват се два брояча, увеличаващ и намаляващ, за да преместим двигателя в диапазон от 0 до 180 градуса.

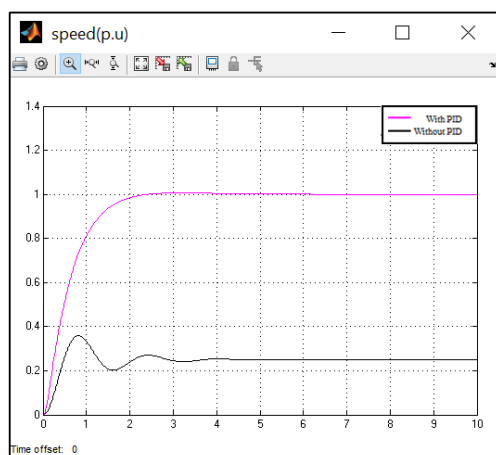
```

ESP32 Servo Motor Demo
2  esp32-servo.ino
3  Uses ESP32Servo Library - https://github.com/madhephaestus/ESP32Servo
4  Uses Seeeduno XIAO ESP32-S3 module
5  Sweeps servo motor 180 degrees in both directions
6  Servo connected to pin 09
7
8
9  DroneBot Workshop 2025
10 https://dronebotworkshop.com
11
12
13 #include <ESP32Servo.h>
14
15 // Create Servo object
16 Servo myservo;
17
18 // Set Servo pin (09 on XIAO ESP32-S3 is GPIO 8)
19 int servoPin = 8;
20
21 void setup() {
22   // Allow allocation of all timers
23   ESP32PWM::allocateTimer(0);
24   ESP32PWM::allocateTimer(1);
25   ESP32PWM::allocateTimer(2);
26   ESP32PWM::allocateTimer(3);
27
28   // Standard 50 Hz servo
29   myservo.setPeriodHertz(50);
30   // Attaches to servo pin using min/max of 500us and 2400us
31   myservo.attach(servoPin, 500, 2400); // Attaches to servo pin using min/max of 500us and 2400us
32 }
33
34 void loop() {
35   for (int pos = 0; pos <= 180; pos += 1) {
36     myservo.write(pos);
37     delay(15);
38   }
39   for (int pos = 180; pos >= 0; pos -= 1) {
40     myservo.write(pos);
41     delay(15);
42   }
43 }
44

```

Фиг. 10. Тестов код за SMC

Симулирана е работата на SMC с MathLab Simulink с параметри $V_S = 5V$, $K_F = 4.6$, $K_P = 1.5$, $K_I = 0.13$, $R = 6,94\Omega$, $L = 0,715mH$ [3]. Регулирането на скоростта с PID и без PID регулатор е показано на фиг. 11.



Фиг. 11. Регулиране скоростта на SMC

ЗАКЛЮЧЕНИЕ

Микро DC серводвигатели намират голямото приложение в роботизирани системи, компютърни устройства и микромашини ЦПУ на DC серводвигатели. От тях се изискват серво операции в реално време, като управление на позицията, управление на скоростта и управление на въртящия момент. Това изисква разработване на система за управление със затворен контур. Изследвани са 32-битови микроконтролери като например ESP32.

Изследването на микроконтролерите и получените симулационни резултати с MathLab Simulink показват много добри резултати при управление на скорост, позиция и въртящ момент.

ЛИТЕРАТУРА

- [1] Nedelchev, T. Prodanov, P. „Simulation and Practical Investigation of a Digital PID Controller in a Servo Drive for DC Motor, 19th Conference on Electrical Machines, Drives and Power Systems (ELMA) 2025.
- [2] Prodanov, P. „Research of Servocontroller for DC Motor Control“, Journal of the Technical University of Gabrovo, Vol. 46'2013 (94-98).
- [3] Abdullah, Z. Shneen, S. „Simulation Model of PID Controller for DC Servo Motor at Variable and Constant Speed by Using MATLAB“, Journal of Robotics and Control Vol. 4, Issue 1, January 2023.
- [4] Shneen, S. Abdullah, Z. „Design and Implementation of No Load, Constant and Variable Load for DC Servo Motor“, Journal of Robotics and Control (JRC) Volume 4, Issue 3, May 2023
- [5] [https://elmchan.org/works/smc/report e.html](https://elmchan.org/works/smc/report%20e.html)
- [6] [https://www.espressif.com/sites/default/files /documentation/esp32_datasheet_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
- [7] <https://www.alldatasheet.com/datasheet-pdf/view/424208/NXP/PCA9685.html>
- [8] <https://dronebotworkshop.com/servoguide/>