УДК 681.5 https://doi.org/10.26160/2541-8637-2023-11-97-100

АВТОМАТИЧЕСКАЯ НАСТРОЙКА ПИ РЕГУЛЯТОРА МЕТОДОМ ГАУССА-ЗАЙДЕЛЯ

Чернов Д.С.
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» имени В.И. Ульянова, Санкт-Петербург

Ключевые слова: регулятор, настройка, ПИ, минимизация, покоординатный спуск, метод Гаусса-Зайделя, оценка качества, переходная характеристика.
Аннотация. В статье рассматривается программа, реализующая автоматическую настройку параметров ПИ регулятора. Решение сводится к автоматической минимизации целевой функции, опирающейся на критерии качества переходного процесса. Программа написана в Matlab, модель объекта управления и блок оценки качества переходной характеристики представлены в графической среде программирования – Simulink.

AUTOMATIC TUNING OF THE PI REGULATOR BY THE GAUSS-SEIDEL METHOD

Chernov D.S.
Saint-Petersburg State Electrotechnical University "LETI", Saint-Petersburg

Keywords: controller, tuning, PI, minimization, coordinate descent, Gauss-Seidel method, quality assessment, transient response.
Abstract. The article discusses a program that implements automatic tuning of the PI controller parameters. The solution is reduced to the automatic minimization of the objective function based on the quality criteria of the transition process. The program is written in Matlab, the model of the control object and the block for evaluating the quality of the transient response are presented in the graphical programming environment – Simulink.

Переходная функция (характеристика) – функция $h(t)$ (её графическое представление), определяющая реакцию системы на единичное ступенчатое воздействие $1(t)$ при нулевых начальных условиях [1]. Встаёт вопрос об оценке качества переходной характеристики(ПХ) системы.

В общем случае, в системах автоматического управления стремятся к тому, чтобы реакция системы (звена) повторяла график управляющего воздействия. Таким образом, задача оценки переходного процесса сводится к минимизации ошибки – разности между задающим воздействием и реальным значением управляемой величины [2]. Ошибка, при условии отсутствия внешних возмущений, возникает из-за инерционности объекта управления, поэтому абсолютного сходства управляющего и реального сигнала на выходе системы достичь проблематично.

Основными критериями качества, характеризующие быстродействие системы являются: время регулирования – время протекающее от момента приложения на вход единичного скачка до момента перехода системы в установившийся режим (ошибка меньше 5%), время нарастания – время, равное отрезку времени, заключенному между точкой пересечения оси времени с касательной, проведённой к осредненной кривой переходной характеристики и

точки пересечения указанной касательной с горизонтальной прямой, соответствующей установившемуся значению управляемой величины[3]. Оба этих критерия можно свести к уменьшению площади под графиком ошибки $e(t)$.

Другим немаловажным критерием оценки качества ПХ является перерегулирование – разность пикового и установившегося значения, делённая на установившееся. Его мы будем учитывать, введя внешнюю штрафную линейную функцию, которая будет срабатывать при превышении графика ПХ прямой, соответствующей установившемуся режиму.

Автоматическая настройка ПИ регулятора представляет собой Matlab скрипт (рис. 1, 2), реализующий минимизацию целевой функции методом Гаусса-Зайделя. Сама целевая функция представляет интеграл от алгебраической суммы двух функций: Основной функции – модуль от ошибки, внешняя штрафная функция – штраф за перерегулирование ПХ.

Модель системы регулирования и блок оценки качества ПХ представлены на рисунке 3.

На рисунке 4 представлена ПХ без ПИ регулятора. Целевая функция оценки качества ПХ в данном случае: 168.2. Вообще говоря, данное значение, при $t \to \infty$, так же стремится к бесконечности из-за наличия статической ошибки. Напомню: мы добиваемся такого результата, чтобы целевая функция стремилась к нулю.

```
clear; clc; close all;
set_param('PI_regulator','SimulationCommand','start');

SimulationTime = 1;
pause(SimulationTime);
EndValue = Output(end,1);
StopSignal = EndValue;
PregulatorValue = str2double(get_param('PI_regulator/Pregulator','Gain'));
IregulatorValue = str2double(get_param('PI_regulator/Iregulator','Gain'));
Step = 0.1;
XorY = 1;
while 1
Stop = 1;
    switch XorY
        case 0
            while 1
                NewValue = PregulatorValue + Step;
                set_param('PI_regulator/Pregulator','Gain',num2str(NewValue))
                set_param('PI_regulator','SimulationCommand','start');
                pause(SimulationTime);
                if abs(Output(end,1)) < abs(EndValue)
                    PregulatorValue = NewValue;
                    EndValue = Output(end,1);
                    Stop = 0;
                    continue;
                else
                    NewValue = PregulatorValue - Step;
                    set_param('PI_regulator/Pregulator','Gain',num2str(NewValue))
                    set_param('PI_regulator','SimulationCommand','start');
                    pause(SimulationTime);
                end
                if abs(Output(end,1)) < abs(EndValue)
                    PregulatorValue = NewValue;
                    EndValue = Output(end,1);
                    Stop = 0;
                    continue;
                else
```

Рис. 1. Первая часть скрипта

```
                    else
                        set_param('PI_regulator/Pregulator','Gain',num2str(PregulatorValue))
                        XorY = 1;
                        break
                    end
                end
            case 1
                while 1
                    NewValue = IregulatorValue + Step;
                    set_param('PI_regulator/Iregulator','Gain',num2str(NewValue))
                    set_param('PI_regulator','SimulationCommand','start');
                    pause(SimulationTime);
                    if abs(Output(end,1)) < abs(EndValue)
                        IregulatorValue = NewValue;
                        EndValue = Output(end,1);
                        Stop = 0;
                        continue;
                    else
                        NewValue = IregulatorValue - Step;
                        set_param('PI_regulator/Iregulator','Gain',num2str(NewValue))
                        set_param('PI_regulator','SimulationCommand','start');
                        pause(SimulationTime);
                    end
                    if abs(Output(end,1)) < abs(EndValue)
                        IregulatorValue = NewValue;
                        EndValue = Output(end,1);
                        Stop = 0;
                        continue;
                    else
                        set_param('PI_regulator/Pregulator','Gain',num2str(IregulatorValue))
                        XorY = 0;
                        break
                    end
                end
        end
        if Stop == 1
            break
        end
    end
```

Рис. 2. Вторая часть скрипта

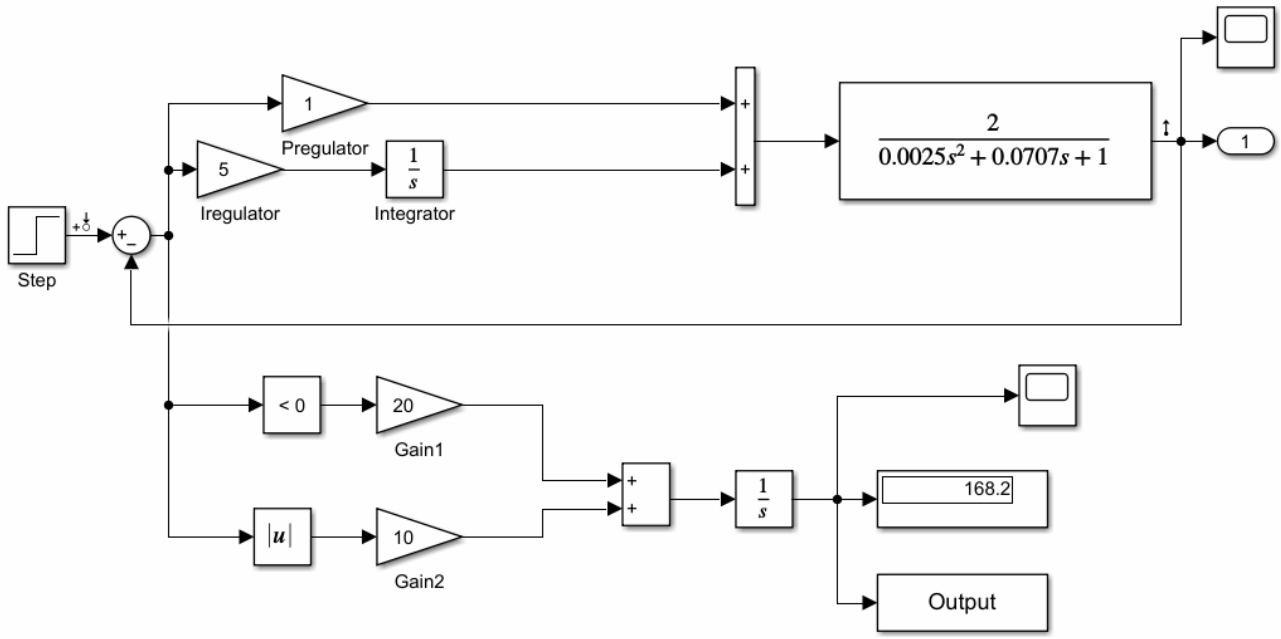

Рис. 3. Модель Simulink

Поставив ПИ регулятор, устанавливаем начальные условия для параметров регулятора (рис. 3): Pregulator, Iregulator. Скрипт автоматически отрегулирует эти параметры в зависимости от целевой функции. Конечный результат представлен на рисунке 5.

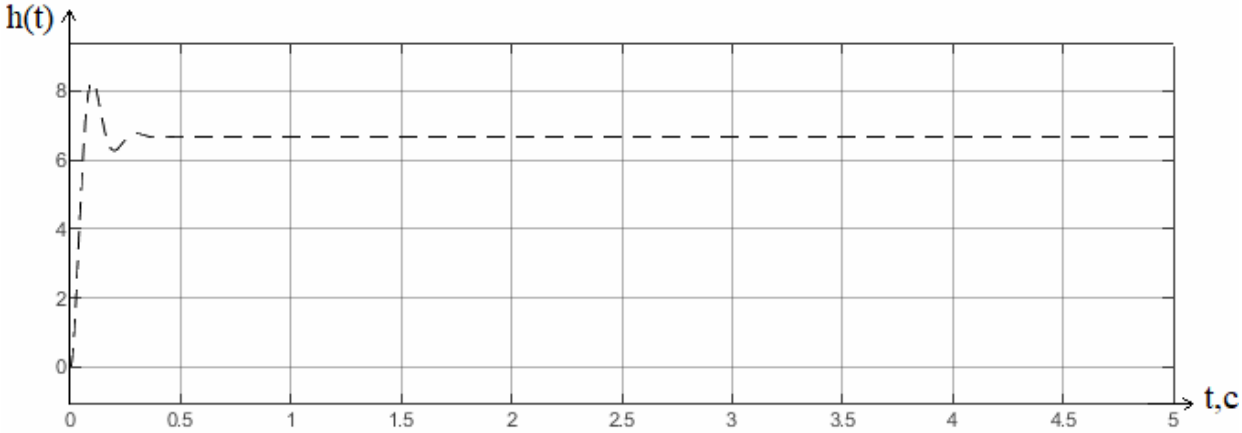

Рис. 4. ПХ системы без ПИ регулятора

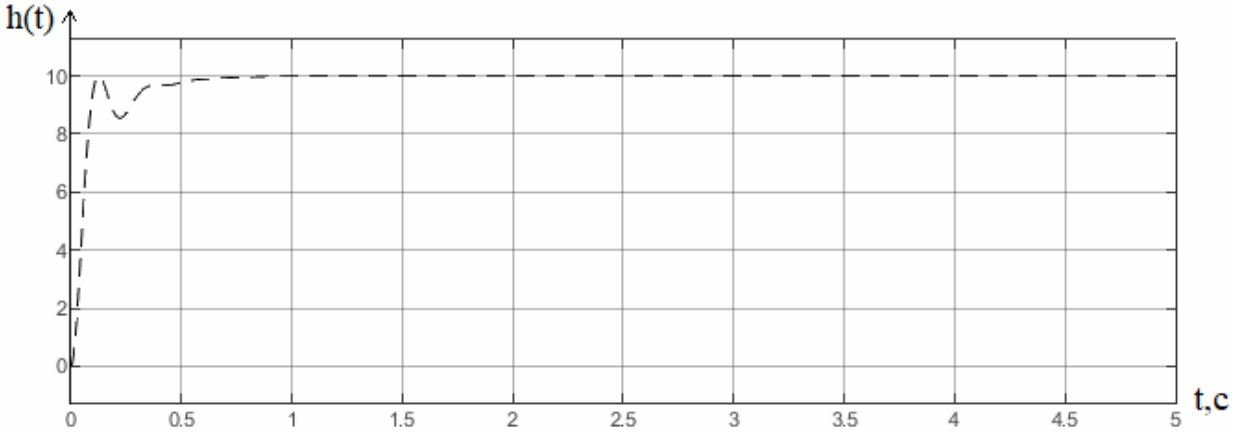

Рис. 5. ПХ после автоматической настройки ПИ регулятора

Перерегулирование отсутствует, статическая ошибка устранена. Значения параметров были установлены Pregulator: 0,75 и Iregulator: 6,15. Значение целевой функции: 8,271 – абсолютный минимум для данного объекта управления. Результат работы программы может разниться в зависимости от шага изменения параметров (в нашем случае 0,05) и значения элементов Gain1 и Gain2 в модели Simulink, которые отвечают за штраф за перерегулирование и модуль ошибки регулирования. Данную программу так же можно использовать для настройки ПД регулятора.

Список литературы

1. Бабаков Н.А., Воронов А.А., Воронова А.А. и др. Теория автоматического управления: Учеб. для вузов по спец. «Автоматика и телемеханика». В 2-х ч. Ч. I. Теория линейных систем автоматического управления; Под ред. А. А. Воронова. – 2-е изд., перераб. и доп. – М.: Высш. шк., 1986. – 367 с.
2. Клавдиев А.А. Теория автоматического управления в примерах и задачах. Ч.I: Учеб. пособие. – СПб.: СЗТУ, 2005. – 74 с.
3. Бесекерский В.А., Попов Е.П. Теория систем автоматического управления. – Изд. 4-е, перераб. и доп. – СПб.: Изд-во «Профессия», 2003. – 752 с.

Сведения об авторе:
Чернов Данил Сергеевич – студент.