

Проектирование нечеткого регулятора следящей системы

© А.А. Бошляков, В.И. Рубцов

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Предложен метод построения нечеткого регулятора для следящего привода. Приведены результаты математического моделирования синтезированной системы.

Ключевые слова: нечеткие системы, ПИД-регулятор, корректирующее устройство.

В теории управления большое внимание уделяется синтезу систем управления при недостаточной информации об объекте управления и действующих на него полезных сигналах и помехах. Один из методов решения этой проблемы – использование нечетких методов управления [1]. В качестве корректирующих устройств систем автоматического управления (САУ) получили широкое распространение пропорционально интегрально-дифференцирующие (ПИД) регуляторы. Настоящая статья посвящена проектированию нечетких ПИД-регуляторов для коррекции следящих систем.

В качестве объекта регулирования выбрали электрический привод на базе двигателя ДП40-20-4-24-Р11-Д40. Была получена математическая модель этого привода и его структурная схема в среде MATLAB Simulink (рис. 1). На основе математического моделирования получены основные характеристики системы, в частности, график переходного процесса (рис. 2).

Была поставлена задача снижения времени переходного процесса. В качестве базовой структуры, формирующей сигнал управления для САУ, выбран адаптивный ПИД-регулятор, изменение коэффициентов которого происходит в зависимости от данных получаемых с нечетко-

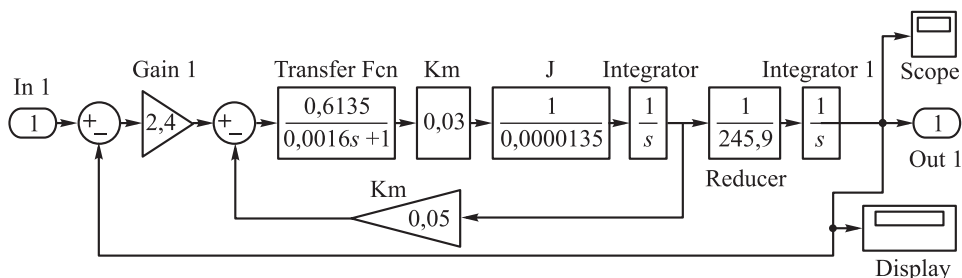


Рис. 1. Структурная схема следящего привода

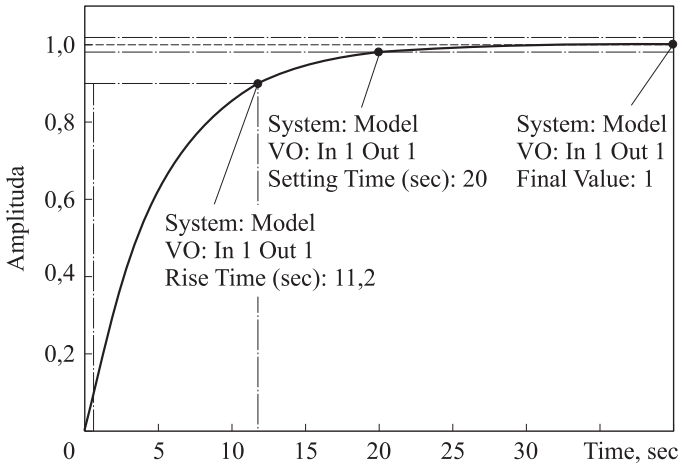


Рис. 2. График переходного процесса системы

го контроллера. Структурная схема нечеткого ПИД-регулятора приведена на рис. 3. Сигнал, подаваемый на силовую часть цепи, состоит из пропорциональной, интегральной и дифференциальной частей. Причем каждая из этих частей регулируется с помощью нечеткой логики в зависимости от двух входных переменных нечеткого блока: ошибки и производной ошибки.

Отличие ПИД-регулятора с контроллером, основанным на нечеткой логике, от обычного заключается в том, что коэффициенты усиления в пропорциональной и интегрирующей цепях регулятора не являются статическими, т. е. зависят от состояния системы в текущий момент времени. Это позволяет качественно изменить процесс управления, учесть параметры сигналов в системе (скорость изменения сигнала, ускорение), а также сделать процесс управления более адаптивным.

В системе управления двигателем постоянного тока по скорости наиболее целесообразно использовать для выбора коэффициентов усиления в регуляторе следующие параметры системы:

- ошибка по углу вращения выходного вала привода редуктора;
- знак скорости вращения выходного вала привода редуктора;

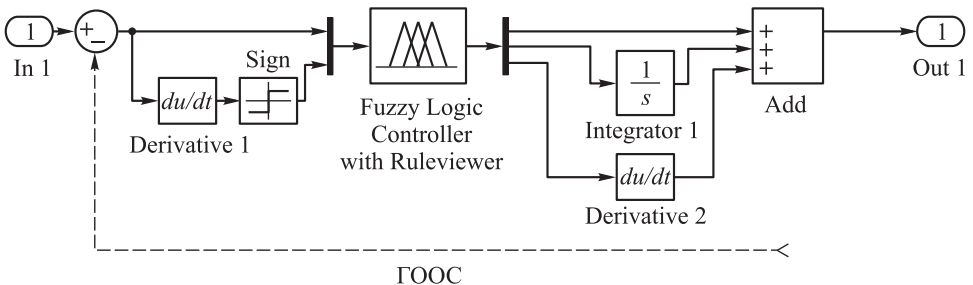


Рис. 3. Структурная схема нечеткого ПИД-регулятора

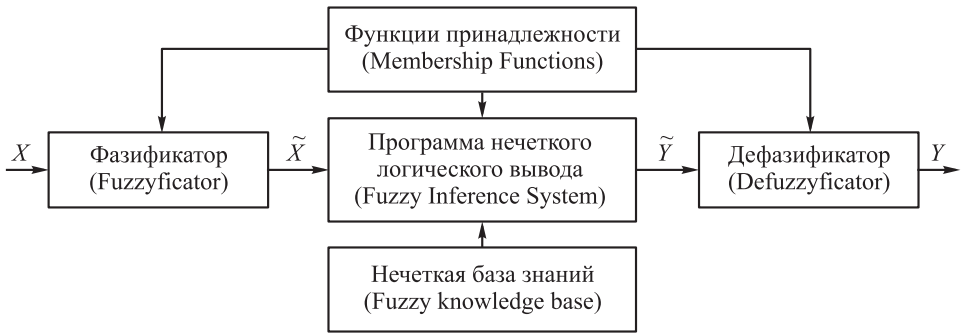


Рис. 4. Структурная схема нечеткого контроллера:

X – входной четкий вектор сигналов, поступающих на контроллер; \tilde{X} – входной нечеткий вектор; \tilde{Y} – выходной нечеткий вектор; Y – выходной четкий вектор сигналов

Структура и принцип работы контроллера, основанного на нечеткой логике. В современных системах применяют контроллеры, основанные на нечеткой логике [2] (рис. 4). Рассмотрим принцип работы контроллера.

1. На вход контроллера поступает необходимое для решения конкретной задачи число входных сигналов (в настоящем проекте два), образующих входной четкий вектор X .
2. Происходит процедура фазификации, т. е. исходя из текущего значения четкого сигнала, на основании известных функций принадлежности каждому сигналу четкого вектора присваивается определенное входное значение (терм), образуя входной нечеткий вектор \tilde{X} . Сигналы, входящие в этот вектор, называются лингвистическими переменными.
3. Программа нечеткого логического вывода (FIS-структура) на основании нечеткой базы знаний ставит в соответствие каждому вектору \tilde{X} выходной нечеткий вектор \tilde{Y} , являющийся результатом нечеткого логического вывода.
4. Значениям (термам) лингвистических переменных, составляющих выходной вектор \tilde{Y} , на основании функций принадлежности ставятся в соответствие определенные четкие значения сигналов, образующие выходной четкий вектор Y , т. е. происходит процедура дефазификации.
5. Сигналы, формирующие вектор Y , поступают в схему регулятора для реализации процесса управления.

Следовательно, первой задачей является определение лингвистических переменных, составление их терм-множеств и функций принадлежности.

Определение логических переменных контроллера и их функций принадлежности. Как было отмечено ранее, входными сигналами для контроллера, основанного на нечеткой логике, являются ошибка по

углу E и знак скорости z_{pi} . Применительно к контроллеру эти сигналы – лингвистические переменные. Для этих переменных вводятся следующие термы:

для лингвистической переменной E

- очень малая (XS);
- малая (S);
- средняя (M);
- большая (L);
- очень большая (XL);

для лингвистической переменной z_{pi}

- отрицательная (Negative);
- нулевая (Zero);
- положительная (Positive).

Поскольку пакет Fuzzy Logic Toolbox не может работать с параметрами, подаваемыми в него извне (в частности, с напряжением U_d), управление будет осуществляться не переменными коэффициентами напряжения, а напряжениями в пропорциональной (U_p) и интегрирующей (U_i) цепях. Этим и обусловлено допущение о работе двигателя в номинальном режиме. На практике при программировании микроконтроллера таких проблем не возникает, и это допущение снимается [3].

Для выходной лингвистической переменной K_p (напряжение в пропорциональной цепи) вводятся термы:

- сильно уменьшенное (ESmall);
- несколько уменьшенное (Small);
- нормальное (Middle);
- несколько увеличенное (Large);
- сильно увеличенное (ELarge);

для лингвистической переменной K_i (напряжение в интегрирующей цепи) – термы

- небольшое прямое (Direct);
- нулевое (Zero);
- небольшое обратное (Inverted);

для лингвистической переменной K_d (напряжение в дифференцирующей цепи) – термы

- нулевое (Zero);
- половина (Half);
- максимальное (Pic).

Для реализации процедур фазификации и дефазификации задаются функции принадлежности для каждой входной и выходной переменных.

Определение функций принадлежности – наиболее трудоемкий процесс, в наибольшей степени определяющий качество процесса управления системой [4]. Так, для составления наиболее адекватной модели требуется не только знание характера поведения системы в целом, но и определенное число экспериментов, позволяющее определить недостатки нечеткой модели и устранить их.

В настоящей статье приведены функции принадлежности уже подвергшиеся экспериментальной оценке и корректировке. Приводить ход этих экспериментов авторы статьи полагают нецелесообразным. Например, для повышения вычислительной способности системы и возможности применения более простых контроллеров функции принадлежности для всех термов выбираются треугольной формы. Применение других форм в такой системе достаточно слабо влияет на результат.

Функции принадлежности для терм-множества лингвистической переменной E представлены на рис. 5, а, для терм-множества лингвистической переменной K_i – на рис. 5, б, для терм-множества лингвистической переменной K_d – на рис. 5, в.

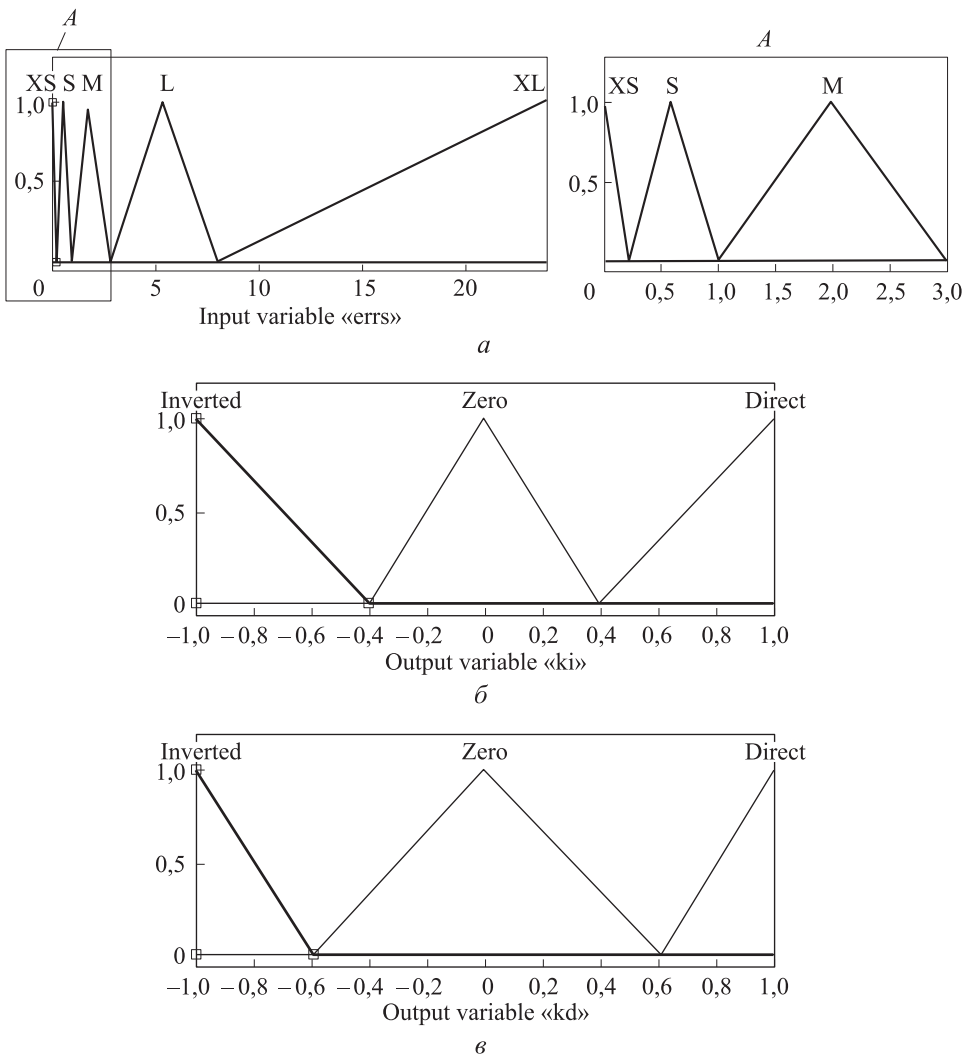


Рис. 5. Функции принадлежности для терм-множества лингвистических переменных E (а), K_i (б) и K_d (в)

1. If (errs is XS) and (znu is Negative) then (k is ESmall)(ki is Inverted)(kd is Zero) (1)
2. If (errs is S) and (znu is Negative) then (k is Small)(ki is Zero)(kd is Half) (1)
3. If (errs is M) and (znu is Negative) then (k is Middle)(ki is Direct)(kd is Zero) (1)
4. If (errs is L) and (znu is Negative) then (k is Large)(ki is Direct)(kd is Zero) (1)
5. If (errs is XL) and (znu is Negative) then (k is ELarge)(ki is Direct)(kd is Zero) (1)
6. If (errs is XS) and (znu is Zero) then (k is ESmall)(ki is Inverted)(kd is Pic) (1)
7. If (errs is S) and (znu is Zero) then (k is Small)(ki is Zero)(kd is Half) (1)
8. If (errs is M) and (znu is Zero) then (k is Middle)(ki is Direct)(kd is Zero) (1)
9. If (errs is L) and (znu is Zero) then (k is ELarge)(ki is Direct)(kd is Zero) (1)
10. If (errs is XL) and (znu is Zero) then (k is ELarge)(ki is Direct)(kd is Zero) (1)
11. If (errs is XS) and (znu is Positive) then (k is ESmall)(ki is Inverted)(kd is Pic) (1)
12. If (errs is S) and (znu is Positive) then (k is Small)(ki is Zero)(kd is Half) (1)
13. If (errs is M) and (znu is Positive) then (k is ELarge)(ki is Direct)(kd is Zero) (1)
14. If (errs is L) and (znu is Positive) then (k is ELarge)(ki is Direct)(kd is Zero) (1)
15. If (errs is XL) and (znu is Positive) then (k is ELarge)(ki is Direct)(kd is Zero) (1)

Рис. 6. База правил

Формирование нечеткой базы знаний контроллера. Под нечеткой базой знаний подразумевается совокупность правил «если – то», определяющих взаимосвязь между входными и выходными лингвистическими переменными контроллера.

В исследуемой системе составляется взаимосвязь между двумя входными (E и znu) и тремя выходными (K_p , K_i и K_d) лингвистическими переменными. При формировании нечеткой базы знаний контроллера также выполняются эксперименты в целях корректировки функций принадлежности лингвистических переменных. В контроллере база знаний задается непосредственно совокупностью правил «если – то» с помощью стандартных функций языков программирования. Взаимосвязь между входными и выходными лингвистическими переменными показана на рис. 6.

Исследуя нечеткую базу знаний контроллера и функции принадлежности лингвистических переменных (см. рис. 5) нетрудно заметить, что в контроллере реализуется идея подачи сигнала с регулятора, несколько опережающего происходящие в системе процессы. Так, при запуске системы коэффициент усиления в пропорциональной цепи повышается, уменьшая время переходного процесса, а при приближении сигнала к требуемому значению резко снижается, уменьшая перерегулирование в системе.

Составление программы нечеткого логического вывода. Программа нечеткого логического вывода представляет собой алгоритм получения нечетких логических заключений, выражающихся в значениях выходных лингвистических переменных. Процедура нечеткого логического вывода происходит на основе нечеткой базы знаний с использованием функций принадлежности лингвистических переменных (см. рис. 4).

Это самая сложная и наиболее требовательная к ресурсам контроллера часть системы нечеткого логического управления. Поэтому при составлении схмотехнической реализации контроллера вопрос о его вычислительной способности должен решаться в первую очередь. Программу нечеткого логического вывода по ходу ее выполнения можно разделить на три этапа (рис. 7). Рассмотрим каждый из этих этапов.

1. Агрегирование подусловий – процедура определения степени истинности по каждому из правил базы знаний системы нечеткого вывода. Если правило состоит из одного условия, то степень его истинности равна значению степени истинности $\beta = x_i = T(i)$ входящего в это условие термина.

Если правило включает в себя несколько условий (сложное правило), то его степень истинности определяется на основе известных значений истинности подусловий. Так, для операции нечеткой конъюнкции (операции «и») двух термов используется формула

$$T(m \text{ и } n) = \min \{T(m); T(n)\}.$$

Например, для правила «если ($E=XS$ И $znu=Negative$) тогда $K=ESmall$ » при значениях $E = 0,1$ и $znu = -1$ его степень истинности равна

$$\beta = \min \{T(XS); T(Negative)\} = \min \{0,545; 1\} = 0,545.$$

2. Активизация подзаключений – процесс нахождения степени истинности каждого из подзаключений правил нечеткой базы знаний. Поскольку в проектируемой системе не вводятся весовые коэффициенты ни для одного из подзакключений, ни для правил в целом, степени истинности каждого из подзакключений будут равны степеням истинности соответствующего правила. После определения всех коэффициентов по каждому из подзакключений, определяются функции принадлежности выходных лингвистических переменных для подзакключений на основании какого-либо метода нечеткой композиции.

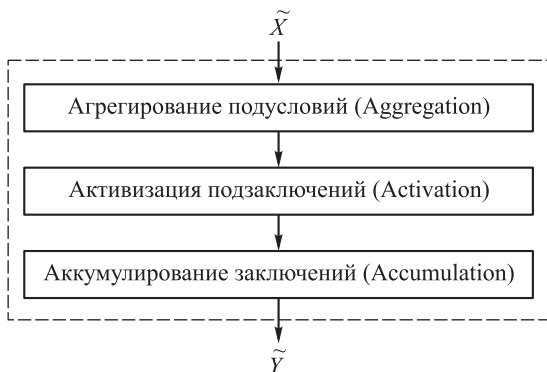


Рис. 7. Основные этапы программы

Для проектируемой системы наиболее целесообразно применить метод так называемой min-активизации, согласно которому функции принадлежности μ'_j для термов j выходных лингвистических переменных определяются по формуле:

$$\mu'_j(Y) = \min \{ \beta', \mu_j(Y) \},$$

где $\mu_j(Y) = T(j)|_Y$ – функция принадлежности четкого выходного сигнала Y .

Например, когда одновременно два терма имеют ненулевую функцию принадлежности, функции принадлежности после активизации для термов и некой выходной лингвистической переменной будут иметь вид, показанный на рис. 8, а и б.

3. Аккумуляция заключений. Цель аккумуляции – объединение всех степеней истинности заключений и подзаключений для получения функций принадлежности каждой из выходных лингвистических переменных.

Необходимость этого этапа связана с тем, что подзаключения, относящиеся к одной и той же лингвистической переменной, могут принадлежать различным правилам нечеткой базы знаний контроллера.

При выполнении аккумуляции заключений последовательно рассматриваются все выходные лингвистические переменные. На полученных в результате активизации коэффициентах β' по каждому из подзаключений для этой переменной (фактически, на основании функций принадлежности μ'_j) формируется функция принадлежности всей выходной лингвистической переменной в целом. Формирование этой функции происходит как объединение функций принадлежности подзаключений, относящихся к рассматриваемой лингвистической переменной:

$$\mu'(Y) = \max \{ \mu'_1(Y), \mu'_2(Y), \dots, \mu'_j(Y), \dots \}.$$

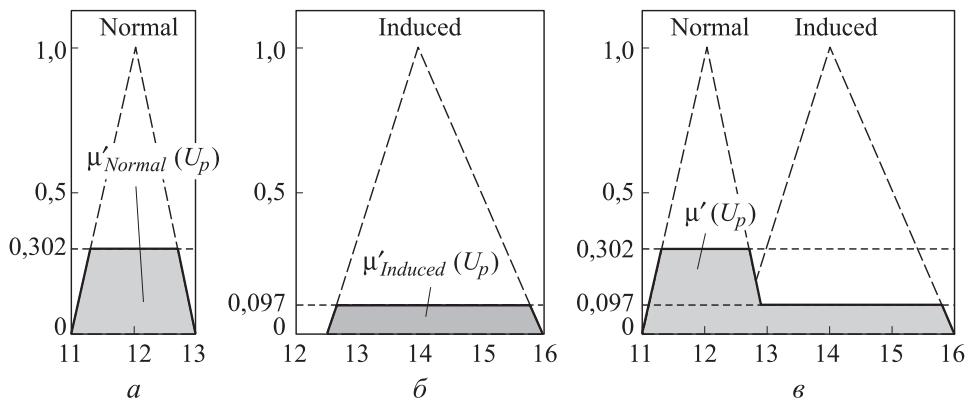


Рис. 8. Функции принадлежности для термов Normal (а), Induced (б) и функция принадлежности после аккумуляции подзаключений (в)

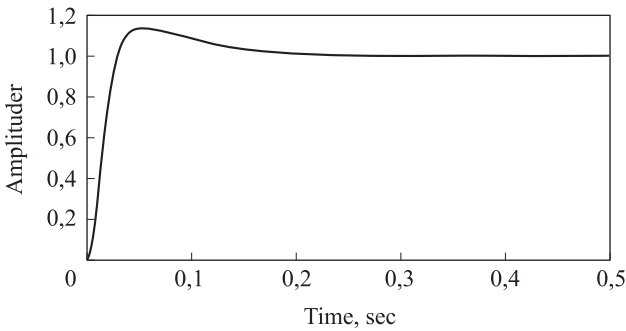


Рис. 10. График переходного процесса системы с нечетким ПИД-регулятором

Например, для указанного выше случая функция принадлежности некой лингвистической переменной после аккумуляции будет иметь вид, показанный на рис. 8, в. Таким образом, формируется программа нечеткого логического вывода.

В результате последующей процедуры дефазификации на основе полученной функции принадлежности формируется выходной четкий сигнал.

Структурная схема скорректированной нечеткой системы приведена на рис. 9. Проводилось моделирование как линейной, так и нелинейной систем. Результаты приведены на рис. 10. Удалось существенно снизить время переходного процесса по сравнению с временем этого же процесса в нескорректированной системе.

Заключение. В настоящей статье предложен метод проектирования нечеткого ПИД-регулятора для коррекции мехатронных приводов. Предложен метод формирования лингвистических переменных при построении нечеткой модели. Разработаны правила формирования нечеткой базы знаний контроллера, программа нечеткого логического вывода. Получена структурная схема скорректированной нечеткой системы. Приведены результаты математического моделирования в среде MATLAB Simulink.

ЛИТЕРАТУРА

- [1] Заде Л. *Понятие лингвистической переменной и его применение для принятия приближенных решений.* Москва, Мир, 1976, 168 с.
- [2] Кудинов Ю.И. Нечеткие системы управления. *Известия Академии наук. Техническая кибернетика*, 1990, № 5, с. 196–206.
- [3] Лохин В.М. *Интеллектуальные системы автоматического управления.* В.М. Лохин, И.М. Макаров ред. Москва, ФИЗМАТЛИТ, 2001, 576 с.
- [4] Захаров В.И., Ульянов В.С. Нечеткие модели интеллектуальных промышленных регуляторов и систем управления: научно-организационные, технико-экономические и прикладные аспекты. *Известия Академии наук. Техническая кибернетика*, 1992, № 5, с. 171–196.

Статья поступила в редакцию 19.07.2013

Ссылку на эту статью просим оформлять следующим образом:

Бошляков А.А., Рубцов В.И. Проектирование нечеткого регулятора следящей системы. *Инженерный журнал: наука и инновации*, 2013, вып. 8.

URL: <http://engjournal.ru/catalog/pribor/robot/936.html>

Бошляков Андрей Анатольевич – канд. техн. наук, доцент кафедры «Специальная робототехника и мехатроника». Автор более 40 научных работ в области проектирования высокоточных мехатронных модулей. e-mail: boshlyakov@mail.ru.

Рубцов Василий Иванович – канд. техн. наук, доцент кафедры «Специальная робототехника и мехатроника». Автор более 20 научных работ в области робототехники.