

В данной работе рассматриваются модели поступательного прямолинейного движения мобильной машины, силового модуля – двигателя и приводится сравнительный анализ управления движением данного объекта с помощью трех контроллеров: НС-нейроконтроллера на основе искусственной нейронной сети, PID-контроллера и контроллера, синтезированного на основе эталонной модели с использованием принципов решения обратных задач динамики.

Литература:

1. Гурский, Н.Н. Моделирование и оптимизация колебаний многоопорных машин: монография / Н.Н. Гурский, Р.И. Фурунжиев. – Минск: БНТУ, 2008. – 296 с.

УДК 681.51.033.26

**Обеспечение заданных показателей качества
интервальной динамической системы**

Несенчук А.А.

Объединенный институт проблем информатики НАН Беларуси

Опишем динамику системы характеристическим полиномом вида

$$p(s) = s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n, \quad (1)$$

где a_j – вещественные коэффициенты, $j = \overline{1, n}$, $s = \sigma + i\omega$

Рассматривается задача размещения семейства корней характеристического уравнения динамической системы порядка n с неопределенностью в заданной области качества Q в плоскости корней s . Область Q ограничивает запас устойчивости системы и задается границами $\beta_1, \beta_2, \beta_{n-2}$ равной степени устойчивости в зависимости от степени полинома (1). С целью вычисления искомым коэффициентов (1), обеспечивающих заданное расположение семейства корней используется расширенный корневой годограф [1] системы. Каждый коэффициент a_j вычисляется на основе j -го уравнения расширения (1), начиная с $j=1$.

Запишем расширение E_n полинома $p(s)$:

$$E_n = \{p_k(s) = s^k + a_1 s^{k-1} + \dots + a_{k-1} s + a_k\}, \quad (2)$$

$$\text{где } k = \overline{1, n}, \quad p_n(s) = p(s) \text{ и } p_{k-1}(s) = p_k(s) - a_k. \quad (3)$$

Выражение (3) является уравнением начальных точек свободного годографа $p_k(s)$. На основе (3) сформулируем следующие утверждения.

Утверждение 1. Корневой годограф полинома $p_{k-1}(s)$ относительно любого из его коэффициентов a_j представляет собой траектории (геометрическое место) начальных точек свободного годографа $p_k(s)$.

Утверждение 2. Если полином $p_{k-1}(s)$, который является порождающим по отношению к полиному $p_k(s)$, асимптотически

устойчив, то все начальные точки свободного корневого годографа $p_k(s)$, за исключением нулевой, располагаются в левой полуплоскости корней.

Следовательно, для настройки полинома (1) с целью обеспечения заданного качества достаточно найти в расширении (2) полинома (1) устойчивый полином степени $k = l$ ближайшей к n и настроить последовательно каждый коэффициент a_j полинома (1) в интервале $l < j \leq n$ посредством настройки свободного члена a_k соответствующего k -го полинома расширения (2), приняв $k = j$. При этом для каждого полинома расширения степени k решается задача устойчивости относительно соответствующей границы $\beta_k = \beta_j$

Литература:

1. Несенчук, А.А. Корневой метод синтеза устойчивых полиномов путем настройки всех коэффициентов / А.А. Несенчук // Автоматика и телемеханика. 2010. № 8. С. 13–24.

УДК 681.142.2

Решение проблемы слабого связывания компонентов в программной среде

Павлунин В. М., Скудняков Ю.А.

Белорусский национальный технический университет

Возрастающая сложность современных программных продуктов требует четкой организации структуры кода приложения, а именно разделение его на переиспользуемые компоненты. При этом между компонентами возникают зависимости, в которых один класс предоставляет услуги, а другой – их потребляет. Чем больше один компонент зависит от другого, тем сложнее в будущем поддерживать их работу, в случае, когда нужно внести изменения в работу зависимого компонента.

Для решения этой проблемы используется такой паттерн проектирования, как внедрение зависимостей. Суть его состоит в том, что зависимые объекты не создаются напрямую классами, их использующими, а достаются из стороннего источника, так называемого контейнера внедрения зависимостей (DI контейнеры). Это сторонние библиотеки, которые предоставляют API для внедрения зависимостей. Для настройки DI контейнеров используются основные три подхода: конфигурационные XML файлы (с расширением `.config`), регистрация с использованием кода и регистрация с использованием конвенций.

Вместе со взятием на себя роли по созданию зависимых компонентов, контейнеры также управляют их жизненным циклом, тем самым устраняя проблемы с утечками памяти, и добиваясь максимально