

АДАПЛАБ-М: ДИРЕКТИВА ДЛЯ САМОНАСТРОЙКИ ИСПЫТАТЕЛЬНОГО СИГНАЛА

А.Г. Александров, Л.С. Михайлова
Институт проблем управления им. В.А. Трапезникова РАН
Россия, 117997, Москва, Профсоюзная, 65
E-mail: alex7@ipu.rssi.ru

Ю.Ф. Орлов

Московский государственный университет им. М.В. Ломоносова
Россия, 119992, Москва, Ленинские горы

1 Введение

В теории идентификации и адаптивного управления можно выделить два направления, зависящие от предположений о внешних возмущениях и помехах измерений. В первом из них возмущения и помехи являются случайными процессами типа "белый шум". Это направление имеет большую историю и во многом связано с методами наименьших квадратов [1]. Во втором направлении возмущения и помехи – неизвестные, ограниченные функции. На объект управления в этом случае подаются испытательные воздействия (испытательные сигналы), и применяются, в частности, методы инструментальных переменных [1], конечно-частотной идентификации [2], рандомизированной идентификации [3].

Естественное ограничение на выбор испытательного сигнала состоит в том, что он должен быть не связан (не коррелирован) с возмущениями и помехами. Во многих случаях это трудно выполнить, так как последние неизвестны. Кроме того, испытательный сигнал не должен заметно влиять на режим нормальной эксплуатации объекта управления и заметно снижать качество управления. Для выполнения этих условий необходима самонастройка испытательного сигнала в реальных условиях. В системе MATLAB программное обеспечение первого направления содержится [4] в пакетах расширения "System Identification Toolbox" и "Frequency Domain Identification". Кроме того, в первом из этих пакетов имеется программное обеспечение метода инструментальных переменных, где предполагается, что испытательный сигнал задается пользователем пакета априори. Такое задание возможно лишь при большом объеме информации о возмущениях, помехах и свойствах объекта.

В работе [5] описано MATLAB-приложение ADAPLAB-M, являющееся программным обеспечением конечно-частотной идентификации и частотного адаптивного управления. Оно состоит из

двух директив¹: "Конечно-частотная идентификация" и "Частотное адаптивное управление".

В методе конечно-частотной идентификации испытательный сигнал имеет определенную структуру: это сумма гармоник, число которых не превышает размерность вектора пространства состояний объекта. Некоррелированность гармоник этого сигнала (с возмущениями и помехами) проверяется в процессе идентификации, когда амплитуды и частоты испытательных гармоник известны. Интуитивно ясно, что испытательные частоты нужно выбирать из диапазона собственных частот объекта, в котором логарифмическая амплитудно-частотная характеристика объекта имеет изломы.

Предлагаемая ниже директива M011 "Самонастройка испытательного сигнала" служит для самонастройки амплитуд и частот этого сигнала в реальных условиях. Функции этой директивы могут быть использованы для планирования эксперимента по применению самонастройки. Для этой цели используется технологическая модель объекта (системы, управляемого процесса, описываемая дифференциальными уравнениями). Она строится на основе знаний специалиста (технолога) по объекту. Это некоторая предполагаемая модель истинного объекта и она может существенно отличаться от истинной его модели. Параметры алгоритма самонастройки, полученные с помощью технологической модели, являются первыми их значениями в реальном эксперименте.

2 Область применения

Рассмотрим асимптотически устойчивый линейный объект, описываемый уравнениями

$$\sum_{i=0}^{\gamma_1} Q^{(i)} s^i q = \sum_{j=0}^{\gamma_2} M^{(j)} s^j u + \sum_{k=0}^{\gamma_3} L^{(k)} s^k f, \quad y = \bar{D}q \quad (1)$$

где $q(t) \in R^n$ – вектор переменных объекта, $y(t) \in R^r$ – измеряемый выход, $u(t) \in R^m$ – испытательный сигнал, $f(t) \in R^\mu$ – внешнее возмущение. Эти уравнения в форме Коши имеют вид

$$\dot{x} = Ax + Bu + Mf, \quad y = Cx + Du, \quad t \geq t_0, \quad (2)$$

¹ Директива – это программа, формирующая интерфейс (ввода исходных данных, вывода промежуточных данных и результатов вычислений), производящая необходимые вычисления (содержащая расчетную часть, позволяющую решать точно описанный класс задач автоматического управления либо идентификации), и формирующая протокол по завершении своей работы.

где $x(t) \in R^n$ – вектор состояния. Матрицы чисел A, B, M, C, D и N неизвестны. Компоненты возмущения – неизвестные, ограниченные функции:

$$|f_i(t)| \leq f_i^* \quad (i = \overline{1, \mu}) \quad (3)$$

где $f_i^* (i = \overline{1, \mu})$ – заданные числа. Выходы и входы объекта не должны превышать заданных границ:

$$|y_i(t)| \leq y_i^*, \quad |u_j(t)| \leq u_j^* \quad i = \overline{1, r} \quad j = \overline{1, m}, \quad (4)$$

где $y_i^*, u_j^* (i = \overline{1, r}, j = \overline{1, m})$ – заданные числа. Числа y_i^* таковы, что выполняется условие

$$|\bar{y}_i(t)| \leq y_i^* \quad i = \overline{1, r}, \quad (5)$$

где $\bar{y}_i(t) (i = \overline{1, r})$ – "естественные" выходы объекта (выходы в режиме его нормальной эксплуатации) когда испытательный сигнал отсутствует ($u=0$).

Элементы передаточной матрицы $W(s) = C(sE - A)^{-1}B + D$ объекта (1) по управлению имеют вид

$$w_{ij}(s) = k_{ij} s^q \frac{\prod_{k=1}^{\bar{p}_{ij}} (\bar{T}_{ij}^{(k)} s + 1) \prod_{k=1}^{\bar{p}_{ij}} (\bar{T}_{ij}^{(k)2} s^2 + 2\bar{T}_{ij}^{(k)} \bar{\xi}_{ij}^{(k)} s + 1)}{\prod_{k=1}^{\bar{p}_{ij}} (\bar{T}_{ij}^{(k)} s + 1) \prod_{k=1}^{\bar{p}_{ij}} (\bar{T}_{ij}^{(k)2} s^2 + 2\bar{T}_{ij}^{(k)} \bar{\xi}_{ij}^{(k)} s + 1)} \quad i = \overline{1, r} \quad j = \overline{1, m} \quad (6)$$

Задача идентификации объекта (либо системы "объект-регулятор") состоит в нахождении оценок постоянных времени передаточных функций (6).

Для идентификации, объект (1) возбуждается испытательными сигналами

$$u_j(t) = \sum_{k=1}^{\rho} \rho_{jk} \sin \omega_{jk} (t - t_o) \quad j = \overline{1, m} \quad (7)$$

чьи амплитуды ρ_{jk} и частоты $\omega_{jk} (j = \overline{1, m}, k = \overline{1, n})$ определяются с помощью предлагаемой директивы M011.

Директива может использоваться для планирования эксперимента по самонастройке испытательного сигнала. В этом случае неизвестные матрицы уравнения (1), возмущение формируется на основе знаний специалиста (технолога) по объекту (либо системе), и тогда модель (1) называется технологической моделью. Ее использование позволяет определить параметры алгоритма самонастройки, которые затем используются при самонастройке в реальных условиях, когда технологическая модель заменяется поступающим в директиву в реальном времени вектором выхода $y(t)$ объекта, возбуждаемого испытательным сигналом (7) и возмущениями.

3 Цели самонастройки

Цель самонастройки испытательных частот состоит в определении их нижней и верхней границ. Это означает, что частоты сигнала (7) должны удовлетворять условиям

$$\omega_{ij}^h \leq \omega_{ij} \leq \omega_{ij}^g \quad (j = \overline{1, m}, k = \overline{1, n}) \quad (8)$$

в которых ω_{ij}^h и ω_{ij}^g – нижние и верхние границы собственных частот передаточных функций $\omega_{ij}(s)$ ($i = \overline{1, r}, j = \overline{1, m}$). Эти границы связаны постоянными времени соответствующих передаточных функций как

$$\omega_{ij}^h = \min \left\{ \frac{1}{\overline{T}_{ij}^{(k1)}}, \frac{1}{\widehat{T}_{ij}^{(k2)}}, \frac{1}{\widetilde{T}_{ij}^{(k3)}}, \frac{1}{\overline{\overline{T}}_{ij}^{(k4)}} \right\}, \quad \omega_{ij}^g = \max \left\{ \frac{1}{\overline{T}_{ij}^{(k1)}}, \frac{1}{\widehat{T}_{ij}^{(k2)}}, \frac{1}{\widetilde{T}_{ij}^{(k3)}}, \frac{1}{\overline{\overline{T}}_{ij}^{(k4)}} \right\} \quad (9)$$

$$i = \overline{1, r}, \quad j = \overline{1, m}, \quad k_1 = \overline{1, \overline{p}_{ij}}, \quad k_2 = \overline{1, \widehat{p}_{ij}}, \quad k_3 = \overline{1, \widetilde{p}_{ij}}, \quad k_4 = \overline{1, \overline{\overline{p}}_{ij}}$$

Для экспериментального определения этих границ, на вход объекта (1) подаются сигналы

$$u_j(t) = \sum_{k=1}^{\rho} \rho_{jk} \sin \omega_{jk}(t - t_0), \quad t_0 + (j-1)\tau \leq t < t_0 + j\tau, \quad j = \overline{1, m}, \quad (10)$$

где τ – заданная длительность j -го эксперимента (время фильтрации в j -ом эксперименте).

Оценки нижних границ находятся по формулам:

$$\omega_{ij}^h(\tau) = \frac{\alpha_{ij}(\tau)\omega}{\beta_{ij}(\tau)} \quad i = \overline{1, r}, \quad j = \overline{1, m} \quad (11)$$

где

$$\alpha_{ij}(\tau) = \frac{2}{\rho_j \tau} \int_{t_0 + (j-1)\tau}^{t_0 + j\tau} y_{ji}(t) \sin \omega_k(t - t_0) dt, \quad (12)$$

$$\beta_{ij}(\tau) = \frac{2}{\rho_j \tau} \int_{t_0 + (j-1)\tau}^{t_0 + j\tau} y_{ji}(t) \cos \omega_k(t - t_0) dt \quad i = \overline{1, r}, \quad j = \overline{1, m}$$

– выходы фильтров Фурье, а ω – заданное достаточно малое число. Оценки верхних границ находятся с помощью похожих формул, где ω – заданное достаточно большое число.

Самонастройка амплитуд ρ_j ($j = \overline{1, m}$) направлена на выполнение требований (4) к границам входов и выходов объекта.

4 Функции самонастройки

Опишем функции самонастройки в рамках ее алгоритма, предложенного в [6]. Пусть $\omega = \omega_o$ – начальная частота – заданное достаточно малое число.

С помощью MATLAB-функции TunRho (определения амплитуд испытательных гармоник) при некотором начальном $\tau = \tau_o$ находятся амплитуды $\rho_j (j = \overline{1, m})$ сигнала (10). Их поиск осуществляется последовательным увеличением либо уменьшением начального значения $\rho_j = u_j^* (j = \overline{1, m})$ с последующим его уточнением половинным делением, так, чтобы выполнялось требование (4) к границам выходов системы.

Эта функция (равно как и приведенные ниже) обращается к MATLAB-функции Analysis (решения уравнений (1) объекта при произвольных входных воздействиях), для которой MATLAB-функциями Test и Dist готовятся векторы значений $u(t)$ – испытательного сигнала и $f(t)$ – внешнего возмущения (с $\eta(t)$ – помехой измерения) на временной сетке $t = t_o + (j-1)\tau : h : t_o + j\tau$, где h – шаг интегрирования дифференциального уравнения (1). Последние три функции отсутствуют, когда директива используется для самонастройки в реальном времени.

Наряду с искомыми амплитудами $\rho_j (j = \overline{1, m})$, MATLAB-функция TunRho вычисляет показатели интенсивности испытательных сигналов

$$\chi_{ij} = \frac{|y_{ij}^{\max} - \bar{y}_{ij}^{\max}|}{y_{ij}^{\max}} \quad i = \overline{1, r}, \quad j = \overline{1, m}, \quad (13)$$

где

$$\begin{aligned} y_{ij}^{\max} &= \max_{t_o + (j-1)\tau + \tau/2 \leq t \leq t_o + j\tau} |y_{ij}(t)| \quad i = \overline{1, r}, \quad j = \overline{1, m} \\ \bar{y}_{ij}^{\max} &= \max_{t_o + (j-1)\tau + \tau/2 \leq t \leq t_o + j\tau} |\bar{y}_{ij}(t)| \end{aligned} \quad (14).$$

На следующем за $\tau = \tau_o$ этапе используется MATLAB-функция FilTimFF (определения времени фильтрации из условия ФФ-фильтруемости возмущения). Время фильтрации $\tau = \tau^*$ определяется ей из неравенств

$$\left| \frac{\bar{\alpha}_{ij}(\tau)}{\alpha_{ij}(\tau)} \right| \leq \varepsilon_{ij}^{\alpha}, \quad \left| \frac{\bar{\beta}_{ij}(\tau)}{\beta_{ij}(\tau)} \right| \leq \varepsilon_{ij}^{\beta} \quad i = \overline{1, r}, \quad j = \overline{1, m} \quad (15)$$

где $\bar{\alpha}_{ij}(\tau)$ и $\bar{\beta}_{ij}(\tau)$ – выходы фильтров Фурье (12), в которых $y_{ij}(t)$ заменено на $\bar{y}_{ij}(t)$, а $\varepsilon_{ij}^{\alpha}$ и ε_{ij}^{β} ($i = \overline{1, r}, j = \overline{1, m}$) – заданные достаточно малые числа.

После выполнения в момент времени $\tau = \tau^*$ неравенств (15), по формуле (11) вычисляются оценки нижних границ собственных частот объекта (1). Процесс фильтрации продолжается теперь уже с использованием MATLAB-функции FilTime (определения времени фильтрации из необходимых условий сходимости границ собственных частот по времени) до тех пор пока не выполняются эти условия:

$$\left| 1 - \frac{\omega_{ij}^H(\tau)}{\omega_{ij}^H(t - \Delta\tau)} \right| \leq \varepsilon_{ij}^\tau, \quad \tau > \tau^* \quad i = \overline{1, r}, \quad j = \overline{1, m} \quad (16)$$

в которых ε_{ij}^τ ($i = \overline{1, r}, j = \overline{1, m}$) - заданные достаточно малые числа.

Затем изложенное повторяется для $\omega = \omega_0 / 2$, где находятся значения $\omega_{ij}^H(\tau^{**})$ при $\tau^{**} > \tau^*$, и проверяются условия

$$\left| 1 - \frac{\omega_{ij}^H(\tau^{***})}{\omega_{ij}^H(\tau^{**})} \right| \leq \varepsilon_{ij}^\omega \quad i = \overline{1, r}, \quad j = \overline{1, m} \quad (17)$$

в которых ε_{ij}^ω ($i = \overline{1, r}, j = \overline{1, m}$) - заданные достаточно малые числа. Если условия (17) выполняются, то искомые оценки

$$\omega_{ij}^H = \omega_{ij}^H(\tau^{***}) \quad i = \overline{1, r}, \quad j = \overline{1, m} \quad (18)$$

В противном случае, в (10) полагается $\omega = \omega_0 / 2^2$, и т.д., до тех пор, пока не выполняются неравенства (17).

Процесс нахождения оценок верхних границ собственных частот аналогичен описанному. При этом $\omega_0 = M \cdot \min_{\substack{1 \leq i \leq r \\ 1 \leq j \leq m}} \{\omega_{ij}^H\}$, где M - достаточно

большое число ($M \approx 10^3$).

Искомые испытательные частоты сигнала (7) находятся с помощью MATLAB-функции TestOm (вычисления испытательных частот). Эти частоты определяются как

$$\log \omega_{jk} = \log \omega_{ij}^H + (k-1) \log \Delta_{ij} \quad i = \overline{1, r}, \quad j = \overline{1, m}, \quad k = \overline{2, n}, \quad (19)$$

где $\log \Delta_{ij} = \frac{\log \omega_{ij}^s - \log \omega_{ij}^H}{n}$ ($i = \overline{1, r}, j = \overline{1, m}$).

Параметрами алгоритма самонастройки являются положительные числа: $\omega_0, \varepsilon_{ij}^\alpha, \varepsilon_{ij}^\beta, \varepsilon_{ij}^\tau, \varepsilon_{ij}^\omega$ ($i = \overline{1, r}, j = \overline{1, m}$) и M .

5 Интерфейс

Директива M011 имеет удобный графический интерфейс, написанный с использованием MATLAB-функций, предназначенных для создания пользовательского интерфейса. Интерфейс запускается с

командной строки системы MATLAB командой DM011. В результате, на экране появляется диалоговое окно директивы, показанное на рис. 1.

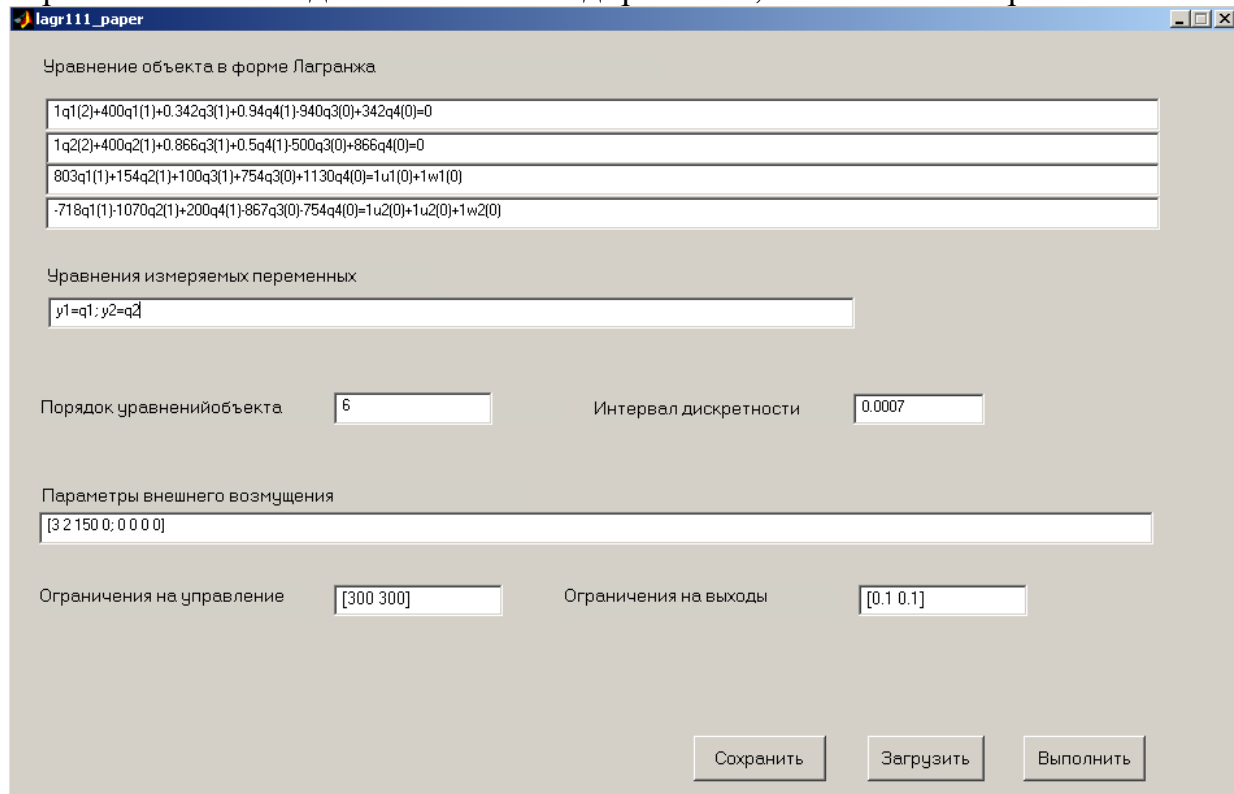


Рис. 1. Интерфейс директивы M011.

Пользователь может ввести исходные данные с клавиатуры, сохранить их в файле для дальнейшего использования, загрузить из файла. При выборе пункта меню "Run" директива запускается на исполнение. В ходе выполнения директивы, на экран выводятся результаты промежуточных вычислений и графики переходных процессов. Результаты работы директивы сохраняются в файле.

6 Пример

Полностью управляемый и полностью наблюдаемый объект - гироскоп [7], описывается системой уравнений вида

$$P \ddot{\beta} + PS\dot{\alpha} + HC\alpha + N\beta = 0, \\ J\dot{\alpha} - C^T HS\alpha - (C^T H + S^T N)\dot{\beta} = Q(u + f), \quad (20)$$

где β_1 и β_2 – углы прецессии (поворота) гироскопов, α_1 и α_2 – проекции абсолютной угловой скорости площадки на ее оси, u_1 и u_2 – моменты двигателей стабилизации (управления). $P = \begin{pmatrix} p_1 & 0 \\ 0 & p_2 \end{pmatrix}$, $Q = \begin{pmatrix} q_1 & 0 \\ 0 & q_2 \end{pmatrix}$,

$$N = \begin{pmatrix} n_1 & 0 \\ 0 & n_2 \end{pmatrix}, \quad H = \begin{pmatrix} h_1 & 0 \\ 0 & h_2 \end{pmatrix}, \quad J = \begin{pmatrix} j_x & 0 \\ 0 & j_y \end{pmatrix}, \quad S = \begin{pmatrix} -\sin \delta_1 & \cos \delta_1 \\ \cos \delta_2 & \sin \delta_2 \end{pmatrix},$$

$$C = \begin{pmatrix} -\cos \delta_1 & -\sin \delta_1 \\ -\sin \delta_2 & \cos \delta_2 \end{pmatrix}, \quad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}.$$

$p_1 = p_2 = 1 \text{ кг} \cdot \text{м}^2$, $q_1 = q_2 = 1$, $n_1 = n_2 = 4 \cdot 10^5 \text{ Н} \cdot \text{мс}$, $h_1 = h_2 = 10^3 \text{ Н} \cdot \text{мс}$, $j_x = 10^2 \text{ кг} \cdot \text{м}/\text{с}^2$, $j_y = 2 \cdot 10^2 \text{ кг} \cdot \text{м}/\text{с}^2$, $\delta_1 = -20^\circ$ и $\delta_2 = 30^\circ$.

Уравнение (20) с этими данными показано на экране интерфейса (рис. 1).

Возмущение $f_1(t) = 150 \text{sign}(\sin 2.1t)$ и $f_2(t) = 0$ и интервал дискретности $h = 0.0007 \text{ с}$.

В результате работа директива M011 по определению нижней границы собственных частот гироплатформы по первому входу (u_1) и первому выходу (y_1) было получено число $\omega_{11} = 7.3$. Правильность этого числа можно проверить аналитически.

7 Заключение

В рамках системы MATLAB предложено программное обеспечение по самонастройке амплитуд и частот испытательного сигнала для моделирования процессов идентификации и адаптивного управления объектами, работающими в условиях неизвестных ограниченных внешних возмущений и помех измерений.

Разработаны MATLAB-функции, используя которые нетрудно сформировать директиву для самонастройки испытательного сигнала в реальных условиях. Для этого в директиве M011 MATLAB-функции Analysis, Test и Dist достаточно заменить файлами экспериментальных данных, получаемых с выхода реального объекта.

Литература

1. *Льюнг Л.* Идентификация систем. Теория для пользователя. -М.: Наука. 1991. 432 С.
2. *Александров А.Г.* Адаптивное управление на основе идентификации частотных характеристик // Известия РАН: "Теория и системы управления". 1995. N 2. С. 63-71.
3. *Граничин О.Н., Поляк Б.Т.* Рандомизированные алгоритмы оценивания и оптимизации при почти произвольных помехах. -М.: Наука. 2003. 291 С.
4. *Дьяконов В.П., Круглов В.* MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник. СПб.: Питер. 2002. 448 С.
5. *Александров А.Г., Орлов Ю.Ф., Михайлова Л.С.* Программное обеспечение конечно-частотной идентификации и адаптивного

управления многомерными объектами // Труды II Международной конференции "Идентификация систем и задачи управления". SICPRO'03. -М.: ИПУ им. В.А. Трапезникова РАН. 2003. CD-ROM N ISBN 5-201-14948-0.

6. Александров А.Г. Конечно-частотная идентификация: самонастройка испытательного сигнала // Труды ЭПИ МИСиС "Робастное управление и частотная идентификация". -Электросталь: ЭПИ МИСиС. 2003. С. 67-97.
7. Александров А.Г. Синтез регуляторов многомерных систем. -М.: Машиностроение. 1986. 272 С.

Приложение. Функции директивы M011

TimeNet – функция формирования временной сетки

Синтаксис:

`t = TimeNet (number, Tdelay, Tfilter, Ndiv, omega)`

Исходные данные:

Number – номер временного интервала,
Tdelay – время задержки (в периодах минимальной частоты),
Tfilter – время фильтрации (в периодах минимальной частоты),
Ndiv – число делений периода максимальной частоты,
omega – частоты ω_k ($k = \overline{1, \zeta}$) испытательного сигнала.

Результаты счета:

t – временная сетка.

Dist – функция внешних возмущений

Синтаксис:

`fun = Dist (par, t)`

Исходные данные:

Par – параметры внешних возмущений:
par(:, 1) – типы n функций $f(t)$ внешних возмущений,
par(:, 1) = 0 – возмущение отсутствует: $f(t) = 0$,
par(:, 1) = 1 – ступенька: $f(t) = \text{par}(:, 2) \cdot \lfloor [t - \text{par}(:, 3)] \rfloor$,
par(:, 1) = 2 – гармоника: $f(t) = \text{par}(:, 2) \cdot \sin[\text{par}(:, 3) \cdot t + \text{par}(:, 4)]$;
t – временная сетка.

Результаты счета:

fun – матрица (n строк) значений внешних возмущений на временной сетке.

Test – функция генератора испытательного сигнала

Синтаксис:

gen = Test (rho, lambda, omega, t)

Исходные данные:

rho – амплитуды ρ_k ($k = \overline{1, g}$) испытательного сигнала,
lambda – декремент λ затухания,
omega – частоты ω_k ($k = \overline{1, g}$) испытательного сигнала,
t – временная сетка.

Результаты счета:

gen – вектор значений испытательного сигнала на временной сетке.

Fourier – функция фильтра Фурье

Синтаксис:

[alf, bet] =
Fourier(rho, lambda, omega, Tdelay, Tfilter, Ndiv, y, t)

Исходные данные:

rho – амплитуды ρ_k ($k = \overline{1, \zeta}$) испытательного сигнала,
lambda – декремент λ затухания,
omega – частоты ω_k ($k = \overline{1, \zeta}$) испытательного сигнала,
Tdelay – время задержки (в периодах минимальной частоты),
Tfilter – время фильтрации (в периодах минимальной частоты),
Ndiv – число делений периода максимальной частоты,
y – фильтруемый сигнал,
t – временная сетка.

Результаты счета:

alf, bet – оценки частотных параметров.

Analysis – функция моделирования при произвольных входных воздействиях

Синтаксис:

[y, x]=Analysis(A, [B M], C, [D N], [u;f], t, x0, flag)

Исходные данные:

A, [B M], C, [D N] – матрицы параметров объекта (1) в форме Коши: $\dot{x} = Ax + Bu + Mf, y = Cx + Du + Nf$,
[u;f] – матрица входных воздействий,
t – временная сетка (вектор времени моделирования),
x0 – начальный вектор состояния,
flag – флаг построения графика:
flag=0 – график строить, flag=1 – график не строить.

Результаты счета:

y – матрица выходных переменных,

x – конечный вектор состояния.

TunRho – функция самонастройки амплитуд испытательного сигнала

Синтаксис:

$[rho, t, x] = \text{TunRho}(A, [B \ M], C, [D \ N], \omega, p1, Ndiv, y_, u_, t, x)$

Исходные данные:

$A, [B \ M], C, [D \ N]$ – матрицы параметров объекта (1) в форме Коши: $\dot{x} = Ax + Bu + Mf, y = Cx + Du + N\eta$,

ω – частота испытательного сигнала,

$p1$ – число периодов минимальной частоты,

$Ndiv$ – число делений периода максимальной частоты,

y_* – ограничение y^* на выход объекта: $|y(t)| \leq y^*$,

u_* – ограничение u^* на вход объекта: $|u(t)| \leq u^*$

t, x – начальное время и вектор состояния.

Результаты счета:

Rho – амплитуда испытательного сигнала,

t, x – конечное время и вектор состояния.

TunOmega – функция определения границ испытательных частот

Синтаксис:

$[bound, t, x] = \text{TunOmega}(A, [B \ M], C, [D \ N], \omega, p1, Ndiv, gam, y_, u_, eps0m, epsT, t, x)$

Исходные данные:

$A, [B \ M], C, [D \ N]$ – матрицы параметров объекта (1) в форме Коши: $\dot{x} = Ax + Bu + Mf, y = Cx + Du + N\eta$,

ω – частота испытательного сигнала,

$p1$ – число периодов минимальной частоты,

$Ndiv$ – число делений периода максимальной частоты,

gam – степень полинома при управлении (для определения нижней границы испытательных частот – gam – любое отрицательное число),

y_* – ограничение y^* на выход объекта: $|y(t)| \leq y^*$,

u_* – ограничение u^* на вход объекта: $|u(t)| \leq u^*$

t, x – начальное время и вектор состояния.

Результаты счета:

$bound$ – нижняя граница (если $gam < 0$) либо верхняя граница (если $gam \geq 0$) испытательных частот,

t, x – конечное время и вектор состояния.

FilTime – функция определения времени фильтрации из необходимых условий сходимости границ собственных частот по времени

Синтаксис:

$[T, Dt, Tend, x] = \text{FilTime}(A, [B \ M], C, [D \ N], \rho, \omega, Tstep, \gamma, n, \epsilon_{T}, Tbegin, x)$

Исходные данные:

$A, [B \ M], C, [D \ N]$ – матрицы параметров объекта (1) в форме Коши: $\dot{x} = Ax + Bu + Mf, \quad y = Cx + Du + N\eta,$

ρ, ω – параметры испытательного сигнала,

$Tstep$ – интервал дискретности,

γ – степень полинома при управлении (для определения нижней границы испытательных частот – γ – любое отрицательное число),

n – число шагов интегрирования,

$\epsilon_{T} - \epsilon_{ij}^{\tau} (i = \overline{1, r}, j = \overline{1, m}),$

t, x – начальное время и вектор состояния.

Результаты счета:

T – время фильтрации,

Dt – текущие значения $\epsilon_{ij}^{\tau} (i = \overline{1, r}, j = \overline{1, m}),$

t, x – конечное время и вектор состояния.

FilTimFF – функция определения времени фильтрации по заданным границам функций фильтруемости

Синтаксис:

$[P1, Tend, x] = \text{FilTimFF}(A, [B \ M], C, [D \ N], \rho, \omega, p1, Ndiv, \epsilon_{Da}, \epsilon_{Db}, Tbegin, x)$

Исходные данные:

$A, [B \ M], C, [D \ N]$ – матрицы параметров объекта (1) в форме Коши: $\dot{x} = Ax + Bu + Mf, \quad y = Cx + Du + N\eta,$

ρ, ω – параметры испытательного сигнала,

$p1$ – число периодов минимальной частоты,

$Ndiv$ – число делений периода максимальной частоты,

$\epsilon_{Da}, \epsilon_{Db}$ – границы функций фильтруемости,

t, x – начальное время и вектор состояния.

Результаты счета:

$P1$ – время фильтрации,

t, x – конечное время и вектор состояния.

TestOm – функция вычисления испытательных частот

Синтаксис:

$om = \text{TestOm}(n, om_{low}, om_{up})$

Исходные данные:

n – порядок объекта (1),

om_{low} – нижняя граница испытательных частот,

om_{up} – верхняя граница испытательных частот.

Результаты счета:

om – набор (n -вектор) испытательных частот.