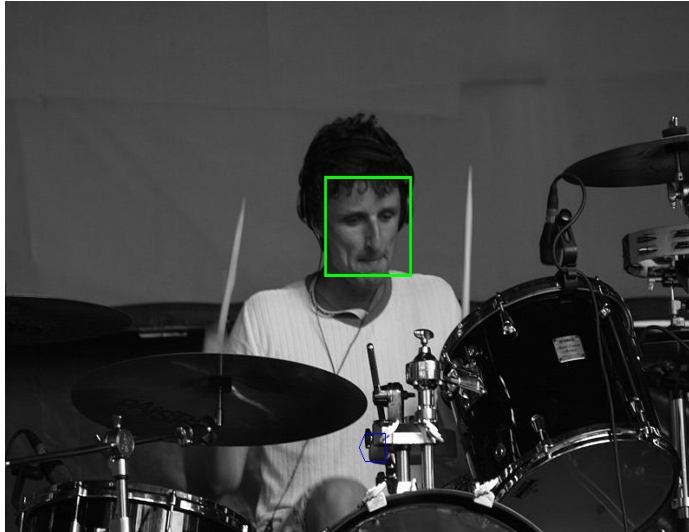


Неабходнасць у распазнанні выяваў узнікае ў самых розных абласцях – ад ваеннай справы і сістэм бяспекі да лічбавання аналагавых сігналаў. Праблема распазнання вобраза набыла выключнае значэнне ва ўмовах інфармацыйных перагрузак, калі чалавек не спраўляецца з лінейна-паслядоўным разуменнем паведамленняў, якія паступаюць да яго, у выніку чаго яго мозг перамякаецца на рэжым адначасовасці ўспрымання і мыслення, якому ўласціва такое распазнаванне.

Прыклад распазнання асобы паказаны на мал. 1.



Мал. 1. Прыклад распазнання асобы

Невыпадкова, такім чынам, што праблема распазнання выяваў апынулася ў полі міждысцыплінарных даследаванняў – у тым ліку ў сувязі з працай па стварэнні штучнага інтэлекту, а стварэнне тэхнічных сістэм распазнання вобраза прыцягвае да сябе ўсе большую ўвагу.

Літаратура

1. Документация OPENCV. – Режим доступа: <https://opencv/docs/>. – Дата доступа: 07.03.2022.
2. Документация dlib. – Режим доступа: <https://dlib.com>.
3. Процессор машинного зрения. – Режим доступа: URL:https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D1%80%D0%B5%D0%BD%D0%B8%D1%8F.

ИССЛЕДОВАНИЕ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ С ПОМОЩЬЮ ПРОГРАММНОГО ПАКЕТА Scilab

К. В. Шкоркина, С. А. Пусев

*Учреждение образования «Гомельский государственный
технический университет имени П. О. Сухого», Республика Беларусь*

Научный руководитель Э. М. Виноградов

Современные вычислительные средства позволяют без особого труда и затрат времени решать сложные задачи управления в технических системах не традиционными инженерными методами, а с использованием математического аппарата любой

степени сложности. При этом не требуется помощь программиста для реализации методов и визуализации проводимых исследований. В настоящее время одним из самых мощных среди универсальных вычислительных пакетов является Matlab. Научные работники и инженеры применяют Matlab для решения задач, возникающих в различных прикладных областях.

Применение Matlab в учебных целях привлекательно, но крайне затруднено из-за его дороговизны. Выходом является использование бесплатного, свободно распространяемого программного продукта Scilab, который предназначен для выполнения инженерных и научных вычислений. Последнюю версию пакета всегда можно скачать с официального сайта программы: www.scilab.org. К сожалению, в настоящее время существует дефицит литературы, посвященной Scilab на языках стран Восточной Европы. В частности, отсутствуют книги на языках стран Восточной Европы для пользователей, желающих использовать Scilab для решения задач автоматического управления.

В инженерной практике наиболее часто используется представление системы автоматического управления в виде структурной схемы, состоящей из типовых динамических звеньев. Каждое звено описывается передаточной функцией. Scilab также позволяет использовать этот подход.

В пакете Scilab имеется функция *syslin()*, которая определяет линейное динамическое звено (или систему) по ее передаточной функции $W(s)$ следующим образом:

$$W = \text{syslin}('c', W(s)),$$

где параметр 'c' указывает, что система с непрерывным временем, а s – символьная переменная.

Затем, используя обычные правила преобразования структурных схем, необходимо получить эквивалентную функцию всей системы управления.

В Scilab имеется функция *csim()*, с помощью которой можно рассчитать временные характеристики системы по ее передаточной функции. Функция *csim()* вычисляет реакцию системы на определенные входные сигналы. Имеется два вида стандартных входных сигналов и соответственно два варианта функции *csim()*:

$$h = \text{csim}('step', t, W),$$

где 'step' – это входной единичный ступенчатый сигнал $1(t)$, а h – это переходная функция $h(t)$;

$$g = \text{csim}('impulse', t, W),$$

где 'impulse' – это входной бесконечный импульс $\delta(t)$ (единичная функция Дирака), а g – это импульсно-переходная функция $g(t)$.

В обоих случаях переменная t – это вектор, определяющий время вычисления функции, а W – передаточная функция системы, определяемая с помощью функции *syslin()*.

В пакете Scilab имеется несколько функций, с помощью которых можно вычислить и построить графики различных частотных характеристик линейных систем, заданных передаточными функциями. Наиболее часто используются следующие функции:

1) *bode()* – диаграмма Боде. Эта функция вычисляет и строит в одном графиче-

ском окне две функции: логарифмическую амплитудно-частотную характеристику (ЛАЧХ) и фазочастотную характеристику (ФЧХ). Функция наиболее часто применяется в формате:

$$bode(W, f_{\min}, f_{\max});$$

2) *nyquist()* – диаграмма (годограф) Найквиста. Эта функция вычисляет и строит график амплитудно-фазовой частотной характеристики (АФЧХ) системы. Функция наиболее часто применяется в формате:

$$nyquist(W, f_{\min}, f_{\max});$$

3) *gainplot()* – функция вычисляет и строит логарифмическую амплитудную частотную характеристику (ЛАЧХ). Функция наиболее часто применяется в формате;

$$gainplot(W, f_{\min}, f_{\max});$$

4) *phaseplot()* – функция вычисляет и строит фазовую частотную характеристику (ФЧХ). Функция наиболее часто применяется в формате:

$$phaseplot(W, f_{\min}, f_{\max}).$$

Во всех этих функциях обозначено f_{\min}, f_{\max} – диапазон частот вычисления в Гц. Возможно применение рассмотренных функций в сокращенном формате:

$$bode(W), nyquist(W), gainplot(W), phaseplot(W).$$

В этом случае по умолчанию минимальная частота $f_{\min} = 10^{-3}$ Гц, а максимальная $f_{\max} = 10^3$ Гц. Этого диапазона частот вполне достаточно для большинства реальных систем.

В Scilab имеется особая часть программ, называемая Xcos, которая позволяет осуществить визуальное математическое моделирование динамических систем различных объектов. Создавать модель в Xcos удобно непосредственно по структурной схеме системы управления. С помощью блочной модели можно исследовать поведение системы при различных видах входных сигналов – ступенчатом, импульсном, гармоническом.

Приведем пример исследования с помощью пакета Scilab системы автоматического управления (САУ), состоящей из трех динамических звеньев. Структурная схема данной САУ приведена на рис. 1.

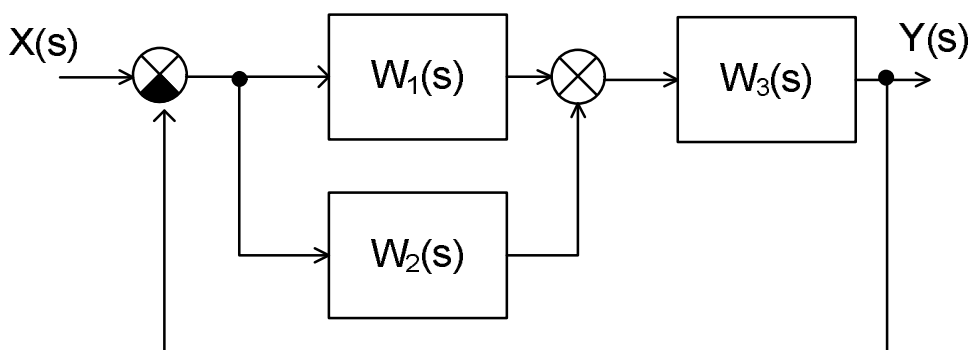


Рис. 1. Структурная схема САУ

Передаточные функции звеньев:

$$W_1(s) = 4; \quad W_2(s) = (0,05s + 1) / (0,1s + 1); \quad W_3(s) = 1 / s(0,5s + 1).$$

Требуется выполнить следующее:

- 1) преобразовать структурную схему и найти эквивалентную передаточную функцию САУ;
- 2) определить устойчивость САУ;
- 3) построить графики переходной и импульсной характеристик.

Для решения поставленных задач создадим несколько скрипт-файлов с помощью редактора SciNotes, входящего в пакет Scilab:

1. Текст скрипта для преобразования структурной схемы САУ и нахождения эквивалентной передаточной функции системы:

```
s = %s; // объявление символьной переменной
// определение передаточных функций отдельных звеньев
W1 = syslin('c', 4, 1);
W2 = syslin('c', 0.05*s + 1, 0.1*s + 1);
W3 = syslin('c', 1, s*(0.5*s + 1));
// преобразование структурной схемы
W12 = W1 + W2;
W123 = W12 * W3;
W = W123 / (1 + W123);
// эквивалентная передаточная функция САУ
disp("Передаточная функция САУ: W = ");
disp(W);
```

Результат выполнения скрипта имеет следующий вид:

$$\text{Передаточная функция САУ: } W = \frac{100 + 9s}{(100 + 29s + 12s^2 + s^3)}$$

2. Текст скрипта для вычисления корней характеристического уравнения САУ с целью определения устойчивости:

```
den = W.den; // вычисление знаменателя передаточной функции САУ
r = roots(den); // вычисление корней характеристического уравнения
disp("Корни характеристического уравнения САУ равны: ");
disp(r);
```

Результат выполнения скрипта имеет следующий вид:

```
Корни характеристического уравнения САУ равны:
-10.109902
-0.9450491 + 2.9996958i
-0.9450491 - 2.999698i
```

Все корни характеристического уравнения имеют отрицательные вещественные части, поэтому можно сделать вывод, что заданная САУ будет устойчивой.

3. Текст скрипта для построения графиков временных функций САУ:

```

t = 0 : 0.01 : 10; // диапазон времени и шаг
h = csim('step', t, W);
plot(t, h); // график функции h(t) в окне с номером 0
xlabel("Переходная характеристика h(t)", "Время, с", "Амплитуда");
xgrid(); // отобразить сетку
scf(1); // открыть новое графическое окно с номером 1
g = csim('impulse', t, W);
plot(t, g); // график функции g(t) в окне с номером 1
xlabel("Импульсно-переходная характеристика g(t)", "Время, с", "Амплитуда");
xgrid(); // отобразить сетку

```

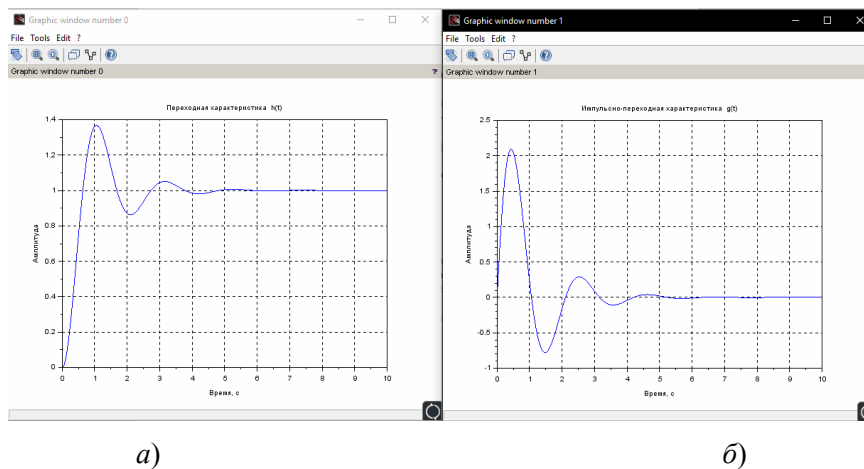


Рис. 2. Графики переходной (а) и импульсной (б) характеристик САУ

Результат выполнения скрипта представлен на рис. 2.

РАЗУМНЫ ДОМ ДЛЯ ІНВАЛІДАЎ ПА ЗРОКУ

М. А. Рогоў, С. І. Моцар

Установа адукацыі «Гомельскі дзяржаўны тэхнічны ўніверсітэт
імя П. В. Сухого, Рэспубліка Беларусь

Навуковы кіраўнік Ю. В. Крышнёў

Згодна з апошнімі данымі ад ААН, усяго ў свеце налічваецца больш за 39 млн цалкам сляпых і 246 млн людзей – з дрэнным зрокам. Абсалютная большасць з іх – гэта людзі, якія па розных прычынах не могуць дазволіць сабе медыцынскае лячэнне. Прычым складаня і часам вельмі дарагія аперацыі даволі часта могуць і не прынесці карыснага эфекту.

Зыходзячы з гэтага, праблемай для грамадства з'яўляецца сацыялізацыя інвалідаў па зроку. Немагчымасць быць самастойным членам чалавечага калектыва, адчуваць сваю інваліднасць і нейкае пачуццё несправядлівасці – гэта штодзённы стрэс амаль кожнага сляпога чалавека.

Вырашэннем праблемы слепаты можа стаць біянічнае вока, якое ўяўляе з сябе біяэлектронную прыладу, што можа вярнуць чалавеку зрок пры поўнай або частковай слепаче. Нягледзячы на даволі аптымістычнае апісанне, яго рэалізацыя натыкаецца на цэлы шэраг разнастайных праблем [1].

Трэба заўважыць, што мы жывем у той час, калі большасць паўсядзённых задач