

Krystyna MACEK-KAMIŃSKA  
Marcin KAMIŃSKI

# ZASTOSOWANIE NARZĘDZI INFORMATYCZNYCH STOSOWANYCH W TWORZENIU STRON INTERNETOWYCH DO OBLICZEŃ SYMULACYJNYCH

**STRESZCZENIE**      *Celem artykułu jest wskazanie możliwości zastosowania intensywnie ostatnio rozwijanych narzędzi informatycznych, zasadniczo przeznaczonych do tworzenia stron internetowych, do obliczeń numerycznych. Głównym obiektem zainteresowania autorów będzie język JavaScript, który w połączeniu z możliwościami języków HTML5 oraz CSS da programistom zestaw wydajnych narzędzi, które wykorzystać będzie można zarówno na etapie obliczeń numerycznych jak i późniejszej prezentacji wyników w formie gotowej strony internetowej.*

**Słowa kluczowe:** języki programowania, metody obliczeniowe  
**DOI:** 10.5604/00326216.1210771

## 1. WSTĘP

---

Jeżeli dokonać analizy stosowanych narzędzi informatycznych w zagadnieniach modelowania i symulacji, to widać wyraźny podział na rosnący udział dedykowanych środowisk obliczeniowych i malejący udział autorskich aplikacji tworzonych w językach programowania. Za popularnością tej pierwszej grupy stoi oczywisty fakt, że wraz z uwzględnianiem w algorytmach obliczeniowych coraz bardziej skomplikowanych modeli coraz trudniej jest indywidualnie przygotować sprawdzoną, wiarygodną aplikację obliczeniową.

---

**dr hab. inż. Krystyna MACEK-KAMIŃSKA, prof. PO**

e-mail: k.macek-kaminska@po.opole.pl

**dr inż. Marcin KAMIŃSKI**

e-mail: m.kaminski@po.opole.pl

Institut Układów Elektromechanicznych i Elektroniki Przemysłowej,  
ul. Próżkowska 76, 45-758 Opole

Sympatie autorów niniejszego artykułu są po stronie tej drugiej grupy, być może dlatego, że programowanie sprawia im dużo satysfakcji.

W ostatnim czasie zostały zaprezentowane doświadczenia autorów z zastosowaniem wybranych nowoczesnych technologii informatycznych w typowych, z punktu widzenia zadań symulacyjnych, obliczeniach. W [1] przedstawiono trzy wersje opracowanej pierwotnie w języku FORTRAN algorytmu estymacji parametrów modelu matematycznego silnika indukcyjnego na podstawie charakterystyk statycznych [2]: wersji mobilnej dla platformy *Android*, aplikacji webowej opartej o ramę programową *Django* oraz aplikacji webowej wykorzystującej język *ActionScript* do zaprogramowania logiki i interakcji z użytkownikiem. Dodatkowo zostały pokazane możliwości stosowanych powszechnie sposobów wymiany danych opartych o standardy XML i JSON.

## 2. JĘZYK PROGRAMOWANIA JAVASCRIPT

---

Język JavaScript został opracowany przez Brendana Eichę dla firmy Netscape do wykorzystania w przeglądarce internetowej Netscape Navigator 2.0. Język ten na tle innych stosowanych powszechnie języków programowania wyróżnia się, zarówno niezwykłą historią powstania (powstał w przeciągu dziesięciu dni w maju 1995 r. [3]) jak i krańcowo odmiennymi opiniami, jakie budzi w środowisku programistów. Pod koniec lat dziewięćdziesiątych XX wieku organizacja ECMA wydała na podstawie JavaScriptu standard języka skryptowego o nazwie ECMAScript. Od tego momentu nazwa JavaScript oznacza jedynie jeden z nadzbiorów języka ECMAScript. Pierwsza wersja standardu była bliska JavaScriptowi w wersji 1.1. Na podstawie tego standardu firma Adobe opracowała język ActionScript wykorzystywany do programowego sterowania animacjami. Ponadto firma Adobe stosuje język JavaScript do sterowania pracą programów Photoshop, Illustrator [4, 5].

Douglas Crockford tak opisuje język: *JavaScript jest językiem ważnym, gdyż jest to język przeglądarek internetowych. Ten związek czyni go jednym z najpopularniejszych języków programowania na świecie. Równocześnie jest to jeden z najbardziej pogardzanych języków. API (ang. Application Programming Interface) przeglądarek – DOM (ang. Document Object Model) – jest nieprzyjazne, a obwiniany za to jest JavaScript. Tymczasem DOM jest niewygodny w każdym języku* [6]. Wspomniany autor, będąc szeroko znanym w świecie autorytetem w dziedzinie języka JavaScript, wskazuje w swojej książce – napisanej z perspektywy kilkunastoletniej historii tego języka – na dobre i złe strony języka JavaScript zdecydowanie koncentrując się na tych dobrych.

Zacytowana wyżej opinia wskazuje na jedną, bardzo znaczącą cechę, jakiej nie mają żadne inne języki programowania opracowane do tej pory. Chodzi o fakt, że każdy współcześnie działający komputer – niezależnie od zainstalowanego na nim systemu operacyjnego – jest w stanie uruchomić kod przygotowany w tym języku. Do tego potrzebna jest jedynie przeglądarka stron internetowych, która z kolei jest podstawowym oprogramowaniem instalowanym w każdym komputerze. Co więcej – możliwe jest też uruchamianie tych kodów na urządzeniach mobilnych – tabletach i smartfonach. Wykorzystując z kolei fakt, że jest to język interpretowany przez silnik przeglądarki – nie jest wymagana instalacja żadnego kompilatora, do tworzenia kodu jest potrzebny tylko edytor tekstu. A to z kolei daje możliwość tworzenia aplikacji nie tylko na komputerze, ale też i na tablecie czy smartfonie, dla których to urządzeń pracujących

pod kontrolą różnych systemów operacyjnych można znaleźć szereg prostych bądź bardziej zaawansowanych edytorów tekstu dla programistów.

Wiele przeglądarek internetowych, poprzez możliwość instalowania specjalnych wtyczek i rozszerzeń (np. narzędzie Firebug dla przeglądarek Firefox/SeaMonkey czy konsola JavaScript dla przeglądarki Chrome), daje wsparcie programistom języka JavaScript.

Kolejną zaletą wynikającą z programowania w języku JavaScript jest fakt, że tak utworzone aplikacje mogą być uruchamiane na komputerze z dostępem do Internetu zlokalizowanym w dowolnym miejscu na Ziemi, co widać na rysunkach 2 i 3.

Ukryta siła języka JavaScript polega na tym, że jest to bardzo zaawansowany język programowania łączący w sobie zarówno cechy języka proceduralnego – co zostanie wykorzystane w dalszej części artykułu – jak i cechy języków obiektowych (co jest cechą często spotykaną w obecnie tworzonych językach programowania) oraz języków funkcyjnych. Cytowany Douglas Crockford tak to podsumowuje: *W głębi JavaScript ma więcej wspólnych cech z językami takimi jak Lisp lub Scheme niż z Javą. Jest to Lisp przebrany za C. Czyni to JavaScript wyjątkowo potężnym językiem* [6].

Ostatnia uwaga każe przyjrzeć się dwóm zdobywającymi coraz większą popularność obiektowo-funkcyjnymi językom programowania uruchamianymi w oparciu o wirtualną maszynę Javy (ang. Java Virtual Machine, w skrócie JVM): Scala i Clojure.

## 2.1. Wykorzystane element języka JavaScript

Jak wspomniano wcześniej, adaptując kod algorytmu estymacji parametrów modelu matematycznego silnika indukcyjnego na podstawie charakterystyki statycznej prądu stojana [4], wykorzystano przede wszystkim proceduralne cechy języka JavaScript. Mając do dyspozycji kod algorytmu estymacji w języku Java [1] jego adaptacja na potrzeby składni języka JavaScript była natychmiastowa. Oba języki mają zbliżoną składnię, korzystają z takich samych struktur danych i instrukcji sterujących. Dodatkowo przygotowany dla aplikacji mobilnej plik danych wejściowych w formacie JSON okazał się natychmiast gotowym do użycia przez język JavaScript – co zresztą nie jest zaskoczeniem, gdyż format JSON (ang. JavaScript Object Notation) jest opracowanym przez cytowanego już Douglasa Crockford alternatywnym do formatu XML tekstowym formatem wymiany danych opartym o podzbiór języka JavaScript. Fragment pliku z danymi wejściowymi w formacie JSON dla omawianej aplikacji przedstawia rysunek 1.

## 2.2. Nowe możliwości wynikające z korzystania z języka JavaScript

Jak zostało wspomniane na wstępie – język JavaScript powstał pierwotnie po to, by w przeglądarce internetowej zapewnić interaktywność poprzez reagowanie na zachodzące zdarzenia. W kontekście omawianej aplikacji do tej pory skupiono się na tych elementach języka programowania JavaScript, które służyły do przeprowadzenia procesu obliczeniowego. Zaletą tego narzędzia jest to, że od razu dostaje się możliwość przygotowania kodu, który również będzie odpowiedzialny za prezentację danych w postaci tabel bądź wykresów na stronie internetowej przygotowanej z wykorzystaniem języka

HTML. Cały kod aplikacji został przygotowany w formacie .html, gdzie w ramach znaczników `<script>...</script>` zawarto cały kod języka JavaScript odpowiedzialny zarówno za część obliczeniową, za interakcję z użytkownikiem jak i za wizualizację danych wejściowych i wyników obliczeń.

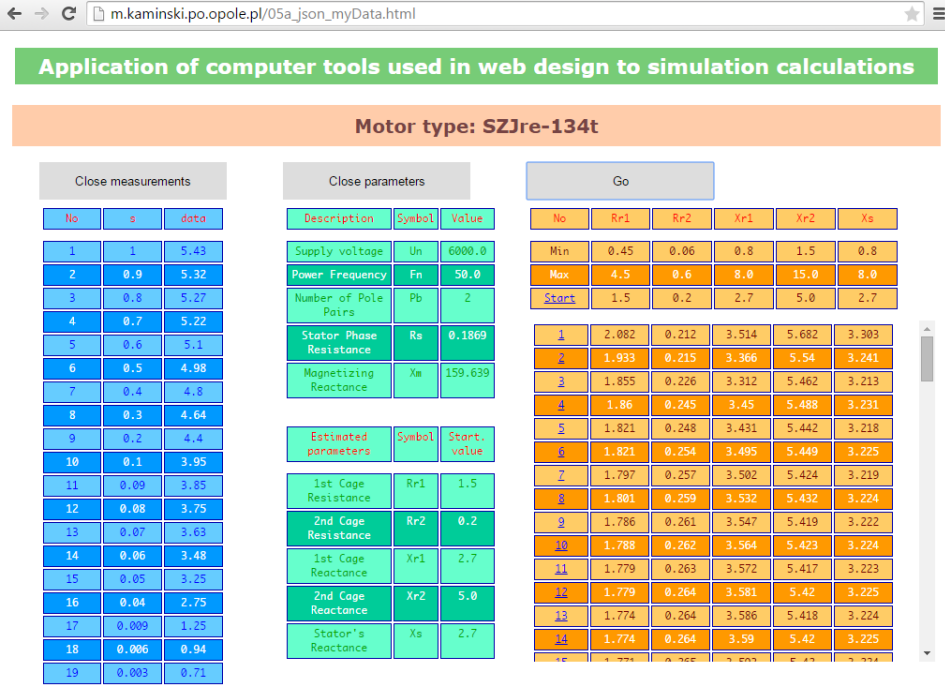
```
var myJSONdata = {
  "data": {
    "motorDc": "1",
    "motorType": "SZJre-134t",
    "numberOfConstantParameters": "5",
    "numberOfEstimatedParameters": "5",
    "myData": {
      "sets": [ { "numberOfMeasurements": "19", "nominalValue": "117.5" } ],
      "constantParameters": [
        { "parameterName": "Supply voltage", "parameterSymbol": "Un", "parameterValue": "6000.0" },
        { "parameterName": "Power Frequency", "parameterSymbol": "Fn", "parameterValue": "50.0" },
        { "parameterName": "Number of Pole Pairs", "parameterSymbol": "Pb", "parameterValue": "2" },
        { "parameterName": "Stator Phase Resistance", "parameterSymbol": "Rs", "parameterValue":
"0.1869" },
        { "parameterName": "Magnetizing Reactance", "parameterSymbol": "Xm", "parameterValue":
"159.639" } ],
      "estimatedParameters": [
        { "parameterName": "1st Cage Resistance", "parameterSymbol": "Rr1", "parameterValueMin":
"0.45", "parameterValueStart": "1.5", "parameterValueMax": "4.5" },
        { "parameterName": "2nd Cage Resistance", "parameterSymbol": "Rr2", "parameterValueMin":
"0.06", "parameterValueStart": "0.2", "parameterValueMax": "0.6" },
        { "parameterName": "1st Cage Reactance", "parameterSymbol": "Xr1", "parameterValueMin": "0.8",
"parameterValueStart": "2.7", "parameterValueMax": "8.0" },
        { "parameterName": "2nd Cage Reactance", "parameterSymbol": "Xr2", "parameterValueMin": "1.5",
"parameterValueStart": "5.0", "parameterValueMax": "15.0" },
        { "parameterName": "Stator's Reactance", "parameterSymbol": "Xs", "parameterValueMin": "0.8",
"parameterValueStart": "2.7", "parameterValueMax": "8.0" } ],
      "measurements": [
        { "characteristicNo": "1", "x": "1", "y": "5.43" },
        { "characteristicNo": "1", "x": "0.9", "y": "5.32" },
        { "characteristicNo": "1", "x": "0.8", "y": "5.27" },
        { "characteristicNo": "1", "x": "0.7", "y": "5.22" },
        { "characteristicNo": "1", "x": "0.6", "y": "5.1" },
        { "characteristicNo": "1", "x": "0.03", "y": "0.71" } ]
    }
  }
}
```

Rys. 1. Treść pliku z danymi do obliczeń w formacie JSON

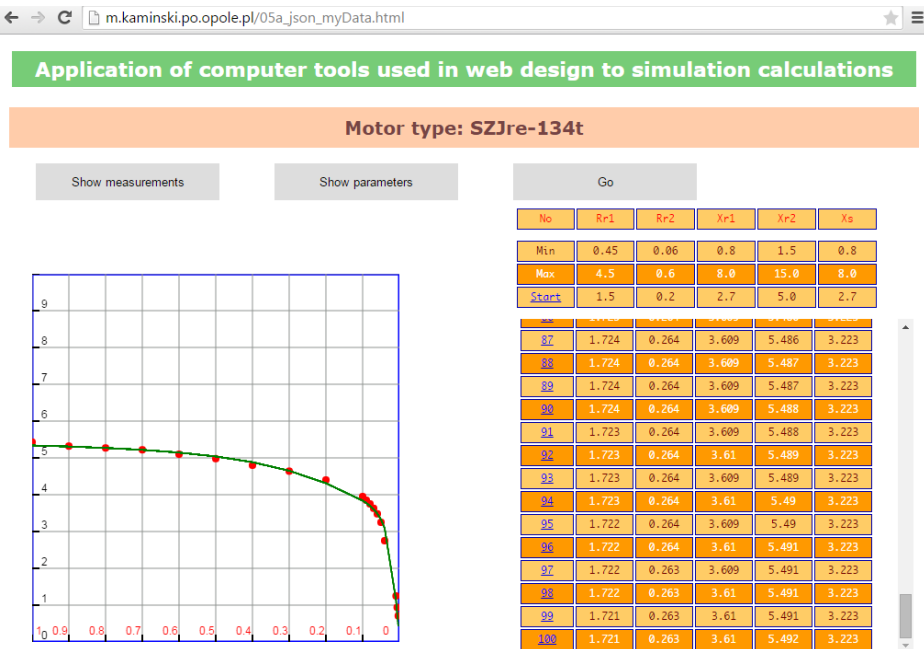
### 3. OPIS APLIKACJI OBLICZENIOWEJ

Rysunek 2 przedstawia stronę internetową aplikacji prezentującą dane pomiarowe i parametry aplikacji oraz wyliczone dane w procesie estymacji. Rysunek 3. prezentuje w formie graficznej wyznaczoną – na podstawie wyestymowanych wartości poszukiwanych parametrów przyjętego modelu matematycznego silnika indukcyjnego – charakterystykę statyczną prądu stojana w funkcji poślizgu (linia zielona) na tle zaznaczonych na czerwono punktów pomiarowych. Kliknięcie myszką na numer iteracji w lewej kolumnie tabeli spowoduje wyznaczenie i pokazanie na wykresie charakterystyki statycznej, obliczonej dla wskazanych wartości estymowanych parametrów modelu, co pokazuje rysunek 3.

Do uzyskania wykresu wykorzystano dostępny w języku HTML od wersji 5. element Canvas, który umożliwia rysowanie grafiki rastrowej na stronach internetowych. Wszystkie operacje, które doprowadziły do utworzenia wykresu (określenie parametrów układu współrzędnych, umieszczenie siatki, wstawienie punktów pomiarowych, narysowanie wykresu) zostały zaprogramowane z poziomu języka JavaScript. W dalszych etapach rozwoju aplikacji możliwe będzie zaprogramowanie interaktywności użytkownika i elementu Canvas.



Rys. 2. Strona internetowa przygotowana w języku JavaScript



Rys. 3. Graficzna prezentacja wyników procesu estymacji

### 3.1. Analiza kodu aplikacji przygotowanej w języku JavaScript

Całość kodu zawarta jest w dwóch plikach. Pierwszy plik jest przygotowany w formacie `.html` i zawiera szkielet kodu strony internetowej, w którym pomiędzy znacznikami `<script>...</script>` zawarty jest cały kod języka JavaScript aplikacji. Drugi niezbędny do działania aplikacji plik to `style.css` – plik zawierający opis formy prezentacji (wyświetlania) stron WWW.

Ilościowa analiza kodu JavaScript aplikacji wskazuje, że około 1/3 całkowitej liczby linii kodu odpowiedzialna jest za przeprowadzanie obliczeń. Pozostała część kodu przeznaczona jest do utworzenia na stronie obiektów typu `Table`, `Button`, wypełnienie ich treścią, przypisaniu działania do utworzonych przycisków, organizację sposobu prezentacji danych na wykresie.

### 3.2. Możliwości rozwoju aplikacji przygotowanej w języku JavaScript

Prezentowane w niniejszym artykule podejście próbuje ustanowić połączenie pomiędzy typowym podejściem do tworzenia aplikacji obliczeniowych opartych o tradycyjne języki programowania (FORTRAN, C/C++/C#) korzystającymi z lokalnych zbiorów z danymi a nowoczesnymi technologiami tworzenia kodu aplikacji opartych o Internet i umożliwiającymi tym aplikacjom korzystanie z zasobów odległych fizycznie baz danych. Wydaje się, że wymóg korzystania ze zdalnych zasobów dostępnych za pomocą połączeń sieciowych będzie już niedługo podstawowym wymogiem funkcjonalnym każdej nowo tworzonej aplikacji.

Istnieje wiele dostępnych rozwiązań w postaci napisanych w języku JavaScript bibliotek pozwalających programistom albo ułatwić sam proces tworzenia kodu albo dodać interesujące rozszerzenia działania aplikacji.

Pierwszym wyborem jest napisana w języku JavaScript biblioteka jQuery [7] pozwalająca, niezależnie od wykorzystywanej przeglądarki, uprościć obsługę elementów struktury obsługiwanego dokumentu z poziomu kodu JavaScript, co skutkować powinno bardziej przejrzystym kodem.

Do wizualizacji otrzymywanych w wyniku obliczeń danych może służyć biblioteka napisana w języku JavaScript do tworzenia dynamicznych i interaktywnych wizualizacji danych na stronach internetowych D3.js (ang. D3 – Data-Driven-Documents) [8].

Kolejną propozycją może być biblioteka Angular.js – jest to wspierana przez firmę Google biblioteka wspierająca programistów w tworzeniu tak zwanych *single page application* SPA w oparciu o sprawdzony i skutecznie wykorzystywany przez wiele narzędzi architekturę MVC (ang. Model-View-Controller) [9].

## LITERATURA

1. Macek-Kamińska K., Kamiński M.: Możliwości nowych języków programowania i ich wykorzystanie w obliczeniach inżynierskich, Ogólnopolska Konferencja Naukowo-Techniczna Postępy w Elektrotechnice Stosowanej PES-8, Kościelisko, 24-28 czerwca, 2013 r.
2. Macek-Kamińska K.: Estimation of the Induction Motor Parameters, Electric Machines and Power Systems, vol. 23, nr 3, s. 329-344, 1995.

3. [https://www.w3.org/community/webed/wiki/A\\_Short\\_History\\_of\\_JavaScript](https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript)
4. Biancuzzi F., Warden S.: Wielkie umysły programowania, Wydawnictwo Helion, Gliwice, 2010.
5. Adobe Systems Incorporated: Adobe Photoshop CS5 – JavaScript Scripting Reference, San Jose, California, 2010.
6. Crockford D.: JavaScript – Mocne strony. Wydawnictwo Helion, Gliwice, 2011.
7. <https://jquery.com/>
8. <http://d3js.org/>
9. <https://angularjs.org/>

Przyjęto do druku dnia 12.05.2016 r.

## APPLICATION OF IT TOOLS USED IN THE CREATION OF WEBSITES TO NUMERICAL CALCULATIONS

Krystyna MACEK-KAMIŃSKA, Marcin KAMIŃSKI

**ABSTRACT** *The aim of the paper is to indicate the possibility of the application of recently intensively developed tools (generally intended to create websites) to numerical calculations. The main object of the authors' interest is the language JavaScript which combined with the capabilities of HTML5 and CSS languages can give developers a set of powerful tools that may be taken both at the stage of numerical calculations and subsequent presentation of results in the form of ready-made website.*

**Keywords:** *programming languages, numerical methods*

**Dr hab. inż. Krystyna MACEK-KAMIŃSKA** – studia na Wydziale Elektrycznym Politechniki Wrocławskiej ukończyła w 1979 r. W 1983 r. obroniła doktorat w Instytucie Układów Elektromaszynowych Politechniki Wrocławskiej. Habilitację uzyskała w 1994 r. w AGH Kraków. Od 1982 r. pracuje w Politechnice Opolskiej (WSI do 1996 r.). W latach 1999-2005 pełniła funkcję prodziekana ds. studenckich wydziału Elektrotechniki i Automatyki. Jest autorem 75 publikacji w tym 2 monografii. Specjalizuje się w badaniach symulacyjnych i eksperymentalnych maszyn elektrycznych, metodach estymacji parametrów, zagadnieniach automatyzacji napędu elektrycznego. Od 2012 roku pełni funkcję prorektora Politechniki Opolskiej ds. dydaktyki.



**Dr inż. Marcin KAMIŃSKI** – studia na Wydziale Elektroniki Politechniki Wrocławskiej ukończył w 1982 r. Od 1983 r. pracuje w Politechnice Opolskiej (WSI do 1996 r.). W 2003 obronił doktorat na Wydziale Elektrotechniki i Automatyki Politechniki Opolskiej. Jest autorem bądź współautorem 30 publikacji. Specjalizuje się w zastosowaniu technik informatycznych w zagadnieniach związanych z modelowaniem i symulacją układów dynamicznych.

