

**STRESZCZENIE ROZPRAWY DOKTORSKIEJ NA TEMAT:  
„Zastosowanie rekurencyjnych sieci neuronowych  
w sterowaniu układem napędowym  
o złożonej części mechanicznej”**

Praca doktorska dotyczy sterowania prędkością napędu elektrycznego ze złożoną częścią mechaniczną. W zadaniu zastosowano rekurencyjne sieci neuronowe. Obiekt, którego istotny element jest reprezentowany jako układ dwumasowy, składa się z dwóch silników połączonych sprężystym wałem. Sprzęgło jest przyczyną drgań skrętnych. W przypadku zastosowania klasycznych struktur sterowania (z regulatorami typu PI) w układzie obserwowane są oscylacje zmiennych stanu. Rezultatem długotrwałego działania, w powyższych warunkach, mogą być uszkodzenia napędu oraz ograniczona dokładność sterowania. Wspomniane zjawiska można powszechnie zaobserwować m.in. w napędach walca-erek, przenośników taśmowych czy teleskopach. Zastosowanie rekurencyjnych sieci neuronowych pozwala zapewnić wysoką dynamikę napędu, a także adaptację do aktualnego stanu napędu.

W ramach rozprawy doktorskiej przedstawiono przegląd literatury związanej z zastosowaniami rekurencyjnych sieci neuronowych w elektrycznych układach napędowych. Zgodnie z aktualnym stanem wiedzy, modele rekurencyjne oferują zwiększoną odporność oraz dynamikę kształtowania sygnałów, w porównaniu z klasycznymi - jednokierunkowymi - strukturami, dlatego zostały wprowadzone jako regulator prędkości napędu ze sprężystym sprzęgłem. Ponadto, w celu wyznaczenia quasi-optimalnych współczynników adaptacyjnych regulatorów rekurencyjnych, zastosowano algorytm metaheurystyczny. Optymalizacja dotyczyła wybranych parametrów, istotnych dla właściwości układu oraz problematycznych w procesie projektowania, obliczano: współczynnik algorytmu adaptacji ( $\eta$ ) oraz wzmocnienia zewnętrzne modelu ( $k_e$ ,  $k_{de}$  i  $k_i$ ). W pracy doktorskiej testowano trzy główne topologie, które zostały zaimplementowane jako przestrajalne regulatory prędkości napędu: sieć Elmana, rekurencyjną sieć falkową (*Recurrent Wavelet Neural Network*) i model typu LSTM (*Long Short-Term Memory*). W etapie optymalizacji zasto-

