

BAB V

KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dari hasil simulasi pengontrolan kecepatan dan posisi kapal selam beserta saran-saran yang bisa dipikirkan kembali untuk menggunakan JST sebagai pengontrol.

5.1 Kesimpulan

Dari hasil pengamatan dan analisis data tugas akhir ini, diperoleh beberapa kesimpulan mengenai pengontrolan kecepatan dan posisi kapal selam menggunakan JST. Beberapa kesimpulan tersebut antara lain :

- Performa dari hasil pengontrolan tidak dapat ditentukan dari nilai *error* yang terjadi pada hasil pengontrolan kecepatan dan posisi karena nilai *error* pada masing-masing variabel yang dikontrol berbeda, melainkan dari performa pelatihan JST. Semakin sedikit jumlah epoch yang dilakukan oleh JST untuk mencapai kriteria *error* yang diinginkan maka semakin cepat waktu yang dibutuhkan oleh pengontrol untuk mengejar target dari pengontrolan kapal selam.
- Performa yang baik dari pelatihan pengontrol JST tidak selalu menghasilkan performa pengontrolan yang terbaik untuk setiap variabel. (sebagai contoh dapat dilihat pada tabel 4.2 . Bandingkan error yang terjadi pada kecepatan di sumbu y pada saat nilai jumlah epoch 12530 dengan nilai jumlah epoch 14134).
- Pada pelatihan pengontrol JST menggunakan arsitektur JST 4 *hidden layer* (30,20,10,5) dengan fungsi aktivasi ‘tansig’ pada *layer* pertama dan kedua

sedangkan pada *layer* ketiga dan keempat menggunakan fungsi aktivasi ‘logsig’, didapatkan performa yang terbaik karena kriteria error yang diinginkan dicapai dengan jumlah pelatihan yang minimum (9848 epoch).

5.2 Saran

Beberapa saran yang dapat diberikan untuk pengembangan penggunaan JST sebagai pengontrol adalah sebagai berikut :

- Karena agak sulit untuk menentukan arsitektur pengontrol JST terbaik, maka akan lebih baik bila jumlah pengontrol JST yang digunakan sebanding dengan jumlah variabel yang dikontrol (*multiple* SISO).
- Pada pelatihan JST sebagai pengontrol nilai bobot awal jaringan lebih baik ditentukan terlebih dahulu, untuk mendapatkan data performa pelatihan JST yang lebih valid.
(Misal : sebuah arsitektur JST dengan parameter A pada pelatihan pertama untuk mencapai kriteria *error* yang diinginkan memerlukan epoch sebanyak N. Sedangkan pada pelatihan kedua untuk mencapai kriteria *error* yang sama memerlukan epoch sebanyak M. Perbedaan jumlah epoch ini mungkin dapat teratasi bila saran di atas direalisasikan).