# THE USE OF ARTIFICIAL INTELLIGENCE IN CONTROLLING A 6DOF MOTION PLATFORM

Webjørn Rekdalsbakken
Institute of Technology and Nautical Science
Aalesund University College
6025 Aalesund, Norway
E-mail: wr@hials.no

**KEYWORDS**
Simulation of Intelligent Systems

## ABSTRACT

Aalesund University College (AUC) has long and broad experience in the design of nautical simulators. An important part of this work is the design and control of motion platforms in three or six Degrees of Freedom (DOF). A small-scale model of a 6DOF motion platform has been designed to investigate the control possibilities of such a device. The platform deck is moved by six actuator rods connected to DC servo motors, controlled by Pulse Width Modulated (PWM) signals. The servo motors are placed in pairs at the corners of an equilateral triangle at the base of the platform, and the actuators are similarly connected in pairs to the platform deck. A solution for the inverse kinematics from platform coordinates to motor shaft angles is derived in this paper. The mathematical transformations are verified through simulations and measurements of the platform movements. To explore the control capabilities of the motion platform, a ball has been controlled with the aid of a camera to follow different kinds of paths on the deck. The ball follows reference paths like a circle or an 8-curve with the platform in a stationary position relative to the camera, as well as while the platform is moving in a plane at a certain height. The control is based on a state space model, and different kinds of adaptive algorithms have been tested, including a Neural Network (NN) and a Genetic Algorithm (GA). The training of the NN controller with a back propagation algorithm was very time demanding, and the real-time implementation was not successful. The GA controller, however, functioned very well and adapted to the different reference conditions in real-time.

## INTRODUCTION

Aalesund University College (AUC) has specialized in the design and building of nautical simulators. The university college is today a leading Norwegian institution in this field with close cooperation to local ship-owners, Rolls Royce Marine Dept. and the Norwegian Marine Technology Research Institute. The simulators are continually upgraded and further developed. The motion of the simulators is based on the octahedral hexapod or Stewart platform. The 6DOF Stewart platform is a much used principle in the construction of all kinds of motion simulators. This platform describes the mathematics of the kinematics between the actuator inputs and the motion of the platform deck. For an historical overview of the development of motion platforms, see (Bonev 2003).[1] AUC has built a small-scale model of a 6DOF motion platform and derived a set of coordinate transformations for the inverse kinematics of the platform. The coordinate transformations have been thoroughly tested on the model. Much experience on design and control strategies had already been obtained from 3DOF motion platforms (Rekdalsbakken 2005) and (Rekdalsbakken 2006). To move the deck one needs six actuators connected in pairs near the edge of the deck at the corners of an equilateral triangle with its centre aligned with the mass centre of the deck. The servo motors are likewise assembled in pairs at the corners of an equilateral triangle at the base of the platform. The motion of each motor shaft is transferred to the deck through a lever arm connected to an actuator arm. A point on the deck is referenced according to a Cartesian coordinate system with the origo at the centre of the platform deck at its rest position and the $z$ -axis in the vertical upright direction. The position of the deck is uniquely determined by the coordinates of the three actuator connection points in this coordinate system, and these coordinates are uniquely defined by a set of motor shaft angles. In the usual operation of the platform the primary signals (roll, pitch, yaw, sway, surge, heave) are available from some kind of source, e.g. a ship simulator. These signals must be transformed to a corresponding set of control signals for the servo drives, giving the reference position for each motor. In this way the platform can be controlled to move in six independent axes. To explore the control capabilities of the platform, several control methods have been tested, see (Rekdalsbakken 2005) and (Bergheim et al. 2006). The moving object on the platform deck represents an open unstable system, and in addition there are inherent nonlinearities in the platform. These factors represent a challenge in the development of stable, robust control strategies for the system, a situation well suited for Artificial Intelligence (AI) methods. Figure 1 shows a picture of the platform model, and Figure 2 shows the geometrical principle of the actuators.

[1] There are many good references on this subject on the Internet, e.g. Wikipedia and ParalleMIC.

Proceedings 21st European Conference on Modelling and Simulation
Ivan Zelinka, Zuzana Oplatková, Alessandra Orsoni ©ECMS 2007
ISBN 978-0-9553018-2-7 / ISBN 978-0-9553018-3-4 (CD)

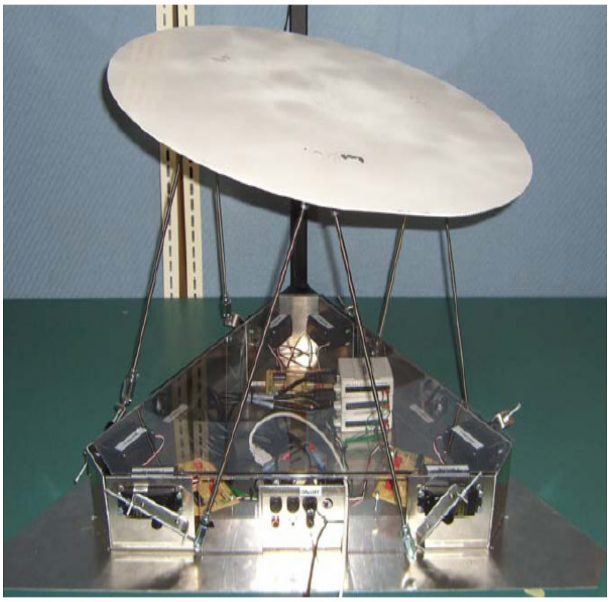

Figure 1: A Picture of the 6DOF Motion Platform

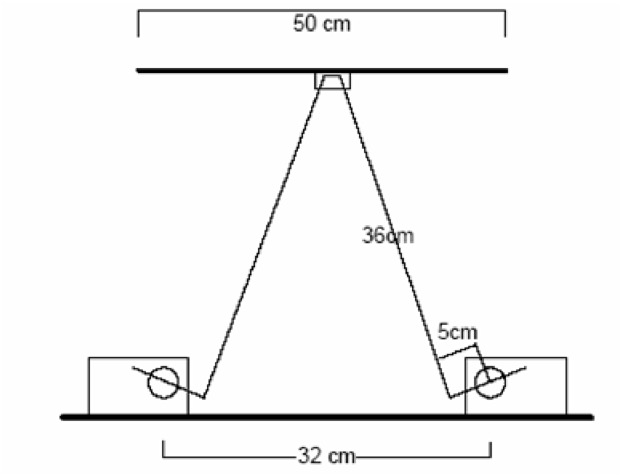


Figure 2: The Geometrical Principle of the Actuators

A further challenge in the control of a 6DOF platform is to make the ball follow a reference path defined relative to the camera position, while the deck itself is moving in the $xy$-plane. If either the ball's reference path or the deck motion is changed on-line, the use of an adaptive controller will be necessary.

## COORDINATES AND TRANSFORMATIONS

### Initial Coordinates and Matrix Transformations

The motion simulator's basic purpose is to move the platform deck from the current position to a new given position. The coordinates of the three connection points between the deck and the actuator arms define the position of the deck. Usually, the input information to the motion platform is the displacement in each of the six axes relative to the initial deck position. The mathematical transformations from these primary signals to the coordinates of the three connection points, is easily described in a 3-dimensional Cartesian coordinate system. This system is defined with the $xy$-plane along the platform deck and the origo at the geometrical centre of the deck in its rest position. In the initial position the deck is horizontal at a given vertical height above the ground, and the $z$-axis is pointing upwards from the deck. With these transformations the new position of the deck can be calculated. This is a straightforward procedure consisting of linear transformations in each of the platform axes. The calculations consist of six subsequent linear operations performed by premultiplying the starting coordinate matrix with the transformation matrix of the respective axis. The result is a 3x3 matrix with columns representing the coordinates of the three connection points in the new position of the deck. This procedure is well documented, see (Rekdalsbakken 2005). To move the deck to its new position the equivalent angle of each of the six servo drives have to be derived. This inverse kinematics from the coordinate matrix to the motor shaft angles is not equally transparent. The calculations are derived in the next section for one of the motor shafts. The procedure is similar for the other motors.

### Calculation of Servo Motor Angle

In the derivation of the motor angle Figure 3 is used as a reference. In this figure point $A(x_1, y_1, z_1)$ is the centre of the motor shaft, point $B(x_a, y_a, z_a)$ is the connection point between the lever arm and the actuator arm, and point $C(x, y, z)$ is the connection point between the actuator arm and the platform deck. In this configuration the points $A$ and $C$ are known, and so are the length $a$ of the lever arm and the length $s$ of the actuator arm. The lever arm is restricted to move in a plane normal to the $xy$-plane in the reference system. This plane forms an angle, $\theta$ with the $x$-axis.

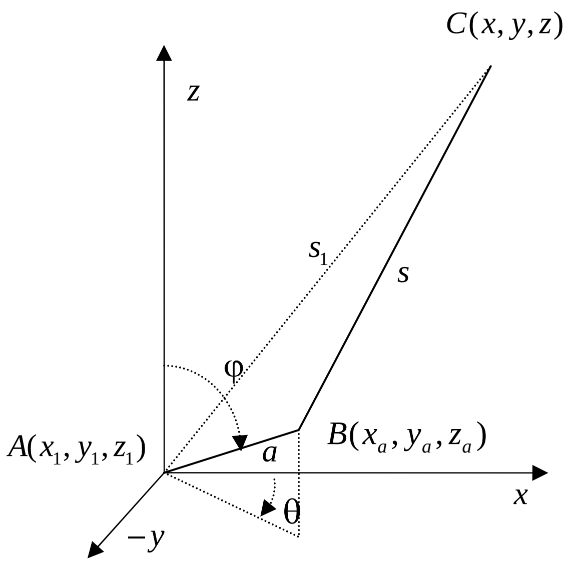


Figure 3: The Geometrical Relation between Lever Arm and Actuator Arm

The actuator arm is connected to the lever arm by a ball joint, so that the actuator arm is free to move in any direction relative to the $xy$-plane. The rest position of the platform deck is chosen such that the angle between the lever arm and actuator arm is 90°. With reference to Figure 3 the $z$-value of the platform deck in this rest position is expressed like this:

$$z_{rest} = \sqrt{s_1^2 - (x - x_1)^2 - (y - y_1)^2} \quad (1)$$

Equation (1) represents the initial height of the platform deck relative to the motor shaft. This initial position is now defined to be the zero $z$-value. A spherical coordinate system is now defined with its origo at the centre of the motor shaft, $A$ and with the length of the lever arm, $a$ as the spherical radius. The lever arm forms the angle, $\varphi$ with the $z$-axis as shown in Figure 3. With reference to this spherical coordinate system the coordinates of the point, $B$ are represented as follows:

$$x_a = a\sin\varphi\cos\theta + x_1 \quad (2)$$

$$y_a = a\sin\varphi\sin\theta + y_1 \quad (3)$$

$$z_a = a\cos\varphi + z_1 \quad (4)$$

The unknown variable is the angle, $\varphi$ which is to be derived. From Figure 3 the following relations are found by using vector arithmetic:

$$s_1^2 = (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 \quad (5)$$

$$a^2 = (x_a - x_1)^2 + (y_a - y_1)^2 + (z_a - z_1)^2 \quad (6)$$

$$s^2 = (x - x_a)^2 + (y - y_a)^2 + (z - z_a)^2 \quad (7)$$

By expanding the equations above and inserting (5) and (6) into (7), the result becomes:

$$\begin{aligned} s^2 = &[s_1^2 - (x_1^2 + y_1^2 + z_1^2) + 2(xx_1 + yy_1 + zz_1)] + \\ &[a^2 - (x_1^2 + y_1^2 + z_1^2) + 2(x_a x_1 + y_a y_1 + z_a z_1)] - \\ &2(xx_a + yy_a + zz_a) \end{aligned} \quad (8)$$

Now expressions (2) through (4) are inserted into (8). The result is:

$$\begin{aligned} s^2 = &s_1^2 + a^2 - 2(x_1^2 + y_1^2 + z_1^2) + \\ &2(xx_1 + yy_1 + zz_1) + \\ &2(x_1 - x)(a\sin\varphi\cos\theta + x_1) + \\ &2(y_1 - y)(a\sin\varphi\sin\theta + y_1) + \\ &2(z_1 - z)(a\cos\varphi + z_1) \end{aligned} \quad (9)$$

The parts of (9) containing the angle, $\varphi$ are now assembled into separate terms. The result becomes:

$$\begin{aligned} s^2 = &s_1^2 + a^2 + \\ &[2(x_1 - x)a\cos\theta + 2(y_1 - y)a\sin\theta]\sin\varphi + \\ &[2(z_1 - z)a]\cos\varphi \end{aligned} \quad (10)$$

Inspection of (10) reveals that the equation represents the trigonometric relationship:

$$M = K\cos\varphi + L\sin\varphi \quad (11)$$

which can also be expressed in the following way:

$$\cos(\varphi - \Psi) = \frac{M}{\sqrt{K^2 + L^2}} \quad (12)$$

Here, $\Psi = \tan^{-1}\frac{L}{K}$ is an auxiliary angle. Comparing (10) with (11) the implied terms are recognized. They are expressed as follows:

$$\begin{aligned} M = &s^2 - s_1^2 - a^2 = \\ &s^2 - a^2 - \left[(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2\right] \end{aligned} \quad (13)$$

$$K = 2(z_1 - z)a \quad (14)$$

$$L = 2(x_1 - x)a\cos\theta + 2(y_1 - y)a\sin\theta \quad (15)$$

With these results the angle, $\varphi$ is derived from (12) and expressed by known quantities like this:

$$\varphi = \cos^{-1}\frac{M}{\sqrt{K^2 + L^2}} + \tan^{-1}\frac{L}{K} \quad (16)$$

The result of the transformation represented by (16) has been confirmed by comparing the calculated angle with measurements on the physical model with the platform deck in several positions referenced to the coordinate system. In addition the results were confirmed by the use of a laser mounted on the platform deck.

## FEEDBACK CONTROL OF A MOVING OBJECT

### Mathematical Model

The basis of the control system is a mathematical model for the dynamics of the moving ball. This model describes the acceleration of the ball for a given angle of the platform deck (Rekdalsbakken 2005). The situation is shown in Fig. 4.

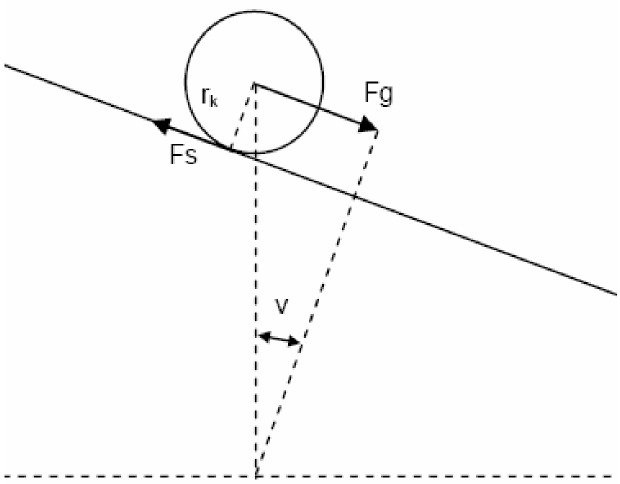


Figure 4: The Dynamics of the Rolling Ball

The torque balance about the centre of the ball combined with the rotational inertia of the ball, $I = \frac{2}{5}mr^2$ gives the following expression for the ball's acceleration:

$$a = \frac{5}{7}\left(g \sin\theta - \dot{\theta}^2 l\right) \tag{17}$$

where $\theta$ is the angle of the platform deck in the given direction, and $l$ is the instant distance from the centre of the deck to the ball. Assuming platform operations within small regions of angles and velocities in the control of the ball, (17) may be linearized and expressed as follows:

$$a \approx \frac{5}{7} g\theta \approx 7.00\theta \tag{18}$$

## Modal Control

Equation (18) is the basis for a simple state space model of the ball dynamics, which is expressed as follows:

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ 7 \end{bmatrix} u$$

where $u = \theta$ is the platform angle. The state variables are the position and speed of the ball in the given direction. A control system can now be designed from this state space model using the control law $u = -K \cdot \vec{x}$ (Ogata 1994). First it is chosen to use two control loops in the independent axes roll and pitch. By using the control law, each loop is expressed by the following model:

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \vec{x} - \begin{bmatrix} 0 \\ 7 \end{bmatrix} \begin{bmatrix} K_1 & K_2 \end{bmatrix} \vec{x} = \begin{bmatrix} 0 & 1 \\ -7K_1 & -7K_2 \end{bmatrix} \vec{x}$$

The control system characteristics are determined by the values of the constants $K_1$ and $K_2$, which are derived from the requirements posed on the control system. To obtain a smooth control of the ball it was decided to use the response of a second order Butterworth polynomial. The bandwidth of the system was stipulated as follows:

$$\omega_b = \frac{2\theta_{max}}{t_r}$$

where $\theta_{max}$ is the maximum platform angle and $t_r$ is the response time. The response time is given by:

$$t_r = \sqrt{\frac{2R}{a}}$$

where $R$ is equal to the radius of the platform, and $a$ is given by (18) inserted $\theta_{max}$. The $K$ matrix was calculated with Matlab, and the control system was tested on the platform.

# CONTROL STRATEGIES

## The Control Problem

To test the motion capabilities of the platform, a relevant and quite demanding experiment is to control the motion of a ball on the platform deck to follow different kinds of reference paths. At first the platform is placed in its rest position and only the two angles roll and pitch are moved. The $K$ matrix derived from the state space model was tuned to make the ball move from a random position to the centre of the deck. The position of the ball is measured with a camera above the platform using a circular search pattern from the deck centre and outwards (Rekdalsbakken 2005). The ball speed is estimated from contiguous measurements of the ball position. When the first goal of centering the ball was achieved, a circle was given as the path of reference for the ball motion. The ball also managed to follow this path. To get a more general and robust control of the ball motion and to extend the experiment to simultaneously move the deck in the $xy$-plane, some AI control methods were explored.

## Neural Network Controller

The purpose of using a NN controller is twofold; first to cope with nonlinearities inherently in the system model and also caused by a non-uniform deck surface, second to achieve a single combined control network with four inputs and two outputs instead of two separate systems. For the use of NN see (Negnevitsky 2002).[2] The neural network has four inputs; the position and speed of the ball in both the roll and pitch directions. The network has two intermediate layers; the first layer has five neurons, the second layer has four. With two hidden layers the network will be capable of dealing with both nonlinear and discontinuous processes. The neuron function is of type sigmoid for all neurons. The network has two outputs representing the new angle of the platform in the roll and pitch axes. The network was trained off-line with the back-propagation algorithm against data obtained from runs with the modal regulator. The data logging and training of the NN became more time-consuming than ecpected.

## Using a Genetic Algorithm

Instead of further testing on the neural network it was decided to introduce a GA, see (Negnevitsky 2002).[3] A very simple chromosome was chosen, consisting only of two genes, representing the two values of the $K$ matrix. Because of the complexity of the calculations it was decided to return to the original system, with two independent and equal control loops. A population of five chromosomes was created. The genes of each

[2] The implementation of NN has been done with the help of Mathworks Neural Network Toolbox, Matlab version 7.0.1 (R14), September 2004.

[3] The implementation of GA has been done with the help of Mathworks Genetic Algorithm and Direct search Toolbox, Matlab version 7.0.1 (R14), September 2004.

chromosome got initial values equal to the modal $K$ matrix superposed a random uncertainty limited to ±10% of the respective $K$-value. Each generation of chromosomes were trained on-line during control of the platform for a given number of cycles, constituting one epoch. The chromosome fitness value was defined by the following optimization criterion:

$$Q = \sum_{i=1}^{n} R - \sqrt{\left(x - x_{ref}\right)^2} \tag{19}$$

where $R$ is a given maximum radius for the area of control on the platform deck and $n$ is the number of cycles in one training epoch. The relative fitness for each of the chromosomes of the current generation is calculated, and these values are used in a roulette wheel technique to select the chromosomes of the next generation. The chromosome with the highest fitness value is transferred unchanged to the next generation, while there is a selection among the others for crossover and mutation. A given probability is set for crossover of a random pair of chromosomes. Likewise there is a certain probability for a chromosome to be exposed to a mutation. The change in a mutation is within 10% of the current gene value. The chromosome and the parameters used in the GA are shown in Table 1. With this GA the chromosomes are updated continuously in real-time. Different kinds of reference paths were trained both with the platform deck in a stationary position and while the deck was moving.

Table 1: Parameters used in the Genetic Algorithm

| NAME | VALUE |
|---|---|
| No. of chrom. in population | 5 |
| No. of genes in chromosome | 2 |
| No. of variables in each gene | 1 |
| Initial value of gene 1 | K1 ± random (0.1K1) |
| Initial value of gene 2 | K2 ± random (0.1K2) |
| Sample time | 30 msec. |
| Epoch time | 500 samples |
| Probability for crossover | 70 % |
| Probability for mutations | 1 % |
| Change in mutation | ±random (0.1K) |

## MANIPULATION AND COMMUNICATION

### Hardware

The servo motors are of the type Futuba S5302 with a voltage supply of 6.0 VDC, and a maximum torque of approx. 1.22 Nm (188.8 oz/inch). Each motor is controlled by a PWM signal from a SG2525A integrated circuit. The control signal to the SG2525A is supplied by a NI-DAQmx USB-6008 with 12 bits analog output resolution. The camera is a Creative Live Ultra web camera with USB connection. It has a video resolution of 640x480 pixels and an acquisition speed of 30 frames pr. sec. Because of the number of USB ports a Trust Compact USB 2.0 hub was used in the experiment assembly. The laboratory setup is shown in Fig. 5.

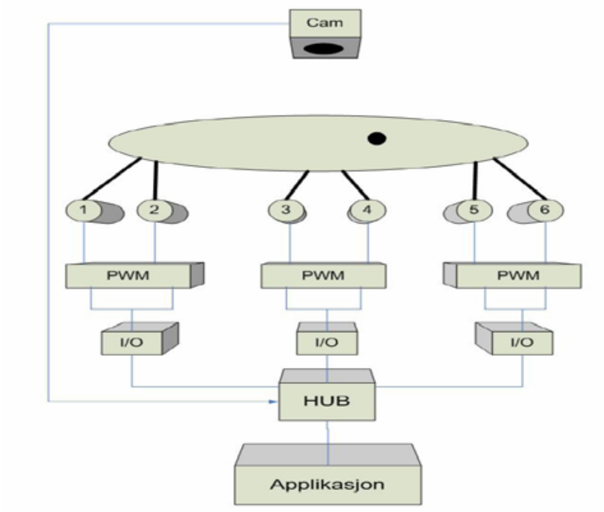


Figure 5: The Laboratory Setup with Actuators and Camera

### Software

The application software was developed with Visual C++ 7.0. The image acquisition and processing were performed with the ActiveX controls VideoOCX and VideoOCX Tools. Proprietary software drivers were used for the camera and I/O cards.

## RESULTS

In spite of comprehensive data logging and training the NN controller never became as good as the modal controller. The idea of training the NN in real-time was abandoned because of the extensive and time-consuming calculations. It might seem as the chosen NN controller was too complicated for adapting to the real-time situation. The GA controller on the other hand functioned very well. This was a reasonably simple algorithm, and the controller consisted of two independent loops, in the roll and pitch directions. When starting with a set of $K$ values tuned for the modal controller, the GA controller showed the ability to adapt in real-time to new types of reference paths. The results of two runs are shown in Figures 6 and 7. In these cases the reference path for the ball is a circle relative to the fixed camera position. Figure 6 shows the case with the modal controller properly tuned for the reference path and the platform deck in the stationary situation. In Figure 7 the GA controller is used, and the deck position is changed from stationary to be circling in the $xy$-plane in the opposite direction of the reference path. The figures show the actual ball motion and also the mean value of the deviation from the reference path. The division of the $x$-axis is in samples (30 msec), and the unit of the $y$-axis is pixels.

## CONCLUSION

The aim of this experiment was to explore the use of adaptive control strategies for a 6DOF motion platform. The development of the control strategies is based on a

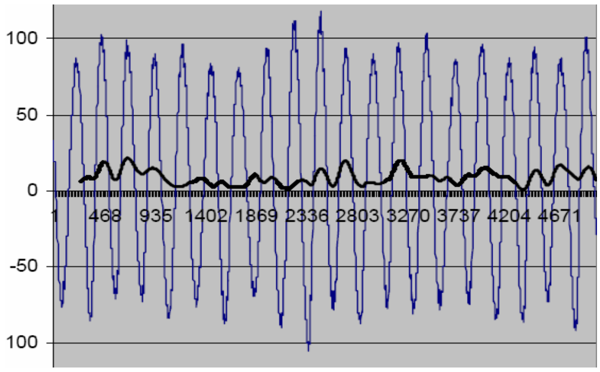


Figure 6: Ball Motion and Deviation of Ball position from the Reference Path with the Modal Controller

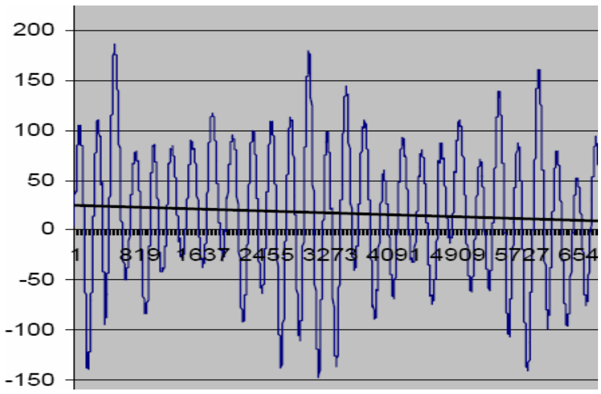


Figure 7: Ball Motion and Deviation of Ball Position from the Reference Path with the GA Controller

modal controller derived from the state space model of the moving ball. Two equal and independent control loops were applied in the independent axes roll and pitch. When properly tuned this control method worked very well both with the platform in the stationary position and in motion in the $xy$-plane. Different paths like a circle and 8-curve were tested with success. However, when there was a change either in the reference path of the ball or in the platform motion, this method got problems with deviations from the reference. To be able to change these conditions in real-time, AI control methods were explored. First a feed forward NN with four inputs and two outputs was tested using the back-propagation training method. With the intention to handle both nonlinearities inherent in the model and discontinuities in the motion, the network was designed with two intermediate layers. The training process, however, became very comprehensive and time-consuming. For this reason the objective of training the network in real-time while controlling the platform was abandoned. The NN controller never became as good as the modal controller. This led to the design of a GA controller. This algorithm is based on the modal controller with two independent control loops. The chromosome population is initialized with values based on the $K$ matrix of the modal regulator. The chromosome population is tested and updated in real-time while the system is running. This method proved to be successful when tested against changes in reference path and in platform motion. The GA adapted the chromosome population to the new situation quite fast, and there seemed to be no problem with the computation time of the algorithm.

## DISCUSSION

The building of this 6DOF motion platform model is part of a quite extensive activity at AUC of designing and building nautical simulators. The simulator research program includes many different fields with vision and motion systems as the basic components. The program is a long-term enterprise in collaboration between AUC and local industry partners including Rolls Royce Marine. Through this project and related laboratory activities AUC can perform small-scale tests of the behaviour of motion platforms. The platform model represents a very suitable tool for exploring different kinds of control strategies, including quite advanced AI techniques. This experience is of great value in the building of true-scale simulators, and also for the understanding of control systems in general. Through this program the scientific staff, too, has gained much experience and new insight.

## REFERENCES

Bonev, I. 2003. “The True Origins of Parallel Robots.” In ParalleMIC – the Parallel Mechanisms Information Center (Jan). http://www.parallemic.org/Reviews/Review007.html.

Bergheim, Ø.; S. Kristiansen and V. Ødegård. 2006. "Intelligent Control of a Moving Ball on a 6DOF Stewart Platform." BSc. thesis in cybernetics for engineering students, AUC.

Negnevitsky, M. 2002. Artificial Intelligence. Pearson Education Limited (2002). Addison-Wesley.

Ogata, K. 1994. Designing Linear Control Systems with MATLAB. Matlab Curriculum Series. PRENTICE HALL (1994).

Rekdalsbakken, W. 2005. "Design and Application of a Motion Platform in Three Degrees of Freedom." In *Proceedings of SIMS 2005, the 46th Conference on Simulation and Modelling*, pp 269-279, Tapir Academic Press, NO-7005 TRONDHEIM.

Rekdalsbakken, W. 2006. “Design and Application of a Motion Platform for a High-Speed Craft Simulator.” In *Proceedings of ICM 2006, IEEE 3rd International Conference on Mechatronics*, pp. 38-43, IEEE Catalog Number of Printed Proceedings: 06EX1432.

## AUTHOR BIBLIOGRAPHY

Webjørn Rekdalsbakken was born in Aalesund, Norway in 1953. He got his MSc. in Physics at The Norwegian Institute of Technology in 1977. He has been a senior engineer in process control at Hydro Aluminium, Rector at Aalesund Engineering College, and now Assoc. Professor at Aalesund University College.