Badminton playing robot - a multidisciplinary test case in Mechatronics ⋆

Julian Stoev PhD. [**,*] **Steven Gillijns PhD.** [***,*]
Andrei Bartic PhD. [****,*] **Wim Symens PhD.** [†,*]

[*] *Flanders Mechatronics Technology Centre, Campus Arenberg, Celestijnenlaan 300D, 3001 Leuven, Belgium*
[**] *e-mail: julian.stoev@fmtc.be*
[***] *e-mail: steven.gillijns@fmtc.be*
[****] *e-mail: andrei.bartic@fmtc.be*
[†] *e-mail: wim.symens@fmtc.be*

Abstract: We present a Mechatronics design approach and related technologies for a badminton playing robot, as a first stage of a multi-year project. The robot is using non-modified shuttles and rackets, which are detected and localized using purely visual information. The robot subsystems are presented: mechanical design, visual detection of the shuttle, shuttle trajectory estimation and interception, actuation, control hardware and software. The paper demonstrates the multidisciplinary nature of the Mechatronics.

Keywords: motion estimation, motor control, optimal trajectory, visual pattern recognition

1. INTRODUCTION

We present a design case for a badminton playing robot. The robot integrates several distinct and inter-operating sub-systems. The sub-systems in question are dealing with visual recognition of the flying shuttle, estimation of its trajectory and possible interception points, moving the robot to the interception point and performing the interception hit.

A particular challenge in the Mechatronics is the integration of diverse technologies from different areas of engineering and science.

Mechanical design of the robot and embedded hard real-time and soft real-time system programming on distributed computing platforms are also an important part of the realization.

Badminton is a very dynamic game and no other robots in this domain are known to the authors. The fastest smash recorded is 332 kph by Chinese mens doubles star Fu Haifeng at the 2005 Sudirman Cup in Beijing.

Due to the high accelerations and velocities of the robot, safety issues are also part of the decision process. As this is a multi-year project, preference was given initially for the safe and tested off-the-shelf technologies and materials, while taking into account the potential for the future by ensuring sufficiently modular design. At the later stages, one can replace the conservatively designed blocks with more aggressive approaches.

Outline of the paper: First the robot concept is presented, including the mechanical configuration. The visual system used for the localization of the flying shuttle is given next. Then we describe our approach for trajectory estimation of the flying shuttle, followed by the motion control part for the robot actuation and discussion on software implementation.

2. MECHANICAL DESIGN

The basic badminton field, the human player, the robot and the flying shuttle are shown on Fig. 1.

The current version of the robot has 3 degrees of freedom:

- **Linear Axis:** – used to position the robot across the field using a linear motor.
- **Rotation Axis:** – used to adjust the position of the racket on the interception plane.
- **Hit Axis:** – used to position the racket for the hit and for performing the actual hit.

The mechanical design of the robot was challenging due to the extremely high acceleration and velocity requirements. The current configuration is the initial design of a multi-year project. Several alternative mechanical configurations were analyzed during the concept phase. This included wheeled robot with omni-wheels and wheel-chair type robots. Some of them were rejected, some of them may be implemented later. The main reason for discarding such options were the very high accelerations and velocities required in the badminton play, which are impossible to achieve in the wheeled robots due to the physical limitations on the wheel traction.

The current mechanical implementation is shown on Fig. 2. The robot in this particular configuration can not cover all

⋆ The project is supported financially by the Flanders government. All project members would like to thank Dirk Wenmakers (dirk.wenmakers@skynet.be) and Dirk Nuyts (Dirk.Nuyts@cit.be) for their valuable contribution as project consultants

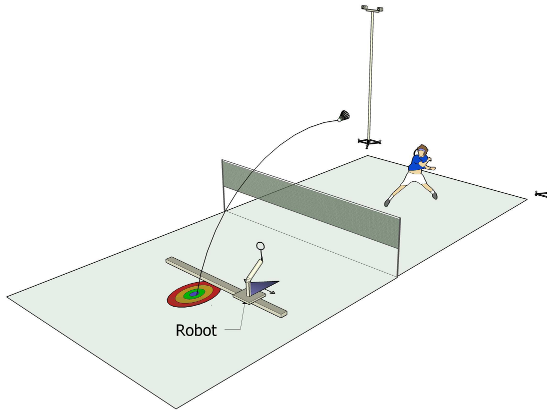

Fig. 1. Concept of the robot playing field and the visual system

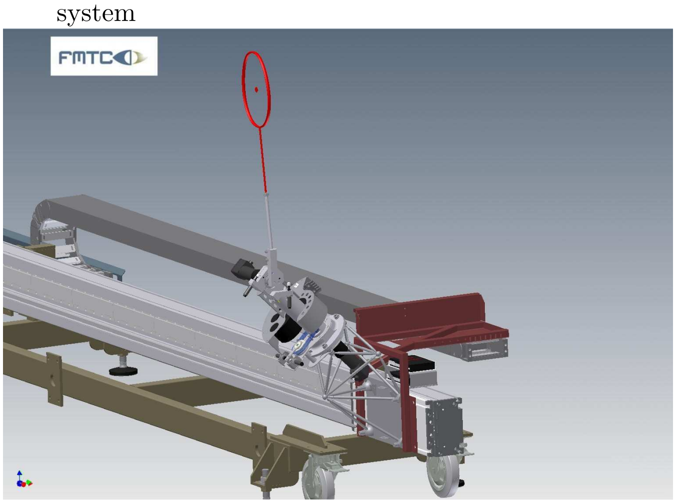

Fig. 2. Robot mechanical design

the field, but this was accepted as a compromise during the first year of the project.

Special effort was made during the design of the mechanical parts to ensure a *clean* slope at the area of the expected cross-over. By clean slope we mean the transfer function shape of the ideal double integrator. The figures 6, 7, 8 show however, that this effort was not entirely successful. The large relatively low frequency vibrating mode of the hitting axis 7 is due to the requirement to use off-the-shelf equipment for the hitting racket and can not be avoided in this case.

Another task during the mechanical design was by appropriately shifting the mass centers to ensure that the axes dynamics are *decoupled* as much, as possible, which would permit the controller design to be performed independently on each axis. This design goal was successful to a very large degree. However there is a remaining coupling between the dynamics of the *hit* axis and the position of the *rotation* axis. This is a nonlinear structural relation, which can not be avoided.

3. THE VISUAL LOCALIZATION SYSTEM

The goal of the localization system is to detect the position of the shuttle. The position will be determined in an absolute 3-D Cartesian coordinate system and will therefore consist of 3 coordinates X, Y, Z and the corresponding time-stamp.

A decision was made that we will not use the so called "image-based visual-servo controller" methods (such as Chesi and Hashimoto (2010)) at this stage despite their apparent simplicity. This is due to the fact that at these methods would have advantages at the close range - when the shuttle is close to the robot, but would not be able to be used effectively at the initial stages of the shuttle trajectory, which is particularly important in our case due to the limited time to move. However such methods are under consideration for the next versions of the robot.

The determination of the 3-D coordinates is based on directional stereo-vision. This method requires that the shuttle is identified in at least 2 images taken from 2 different positions. It then makes a transformation from 2-D pixel coordinates to 3-D world coordinates. The two images can be taken by for instance two video cameras. The optical axis of the 2 cameras must not be in parallel.

The visual system has been designed starting from the requirements set by the rules of the badminton game.

- The field of view is such that it can cover a full badminton field of 13.4m by 6.1m from 2m behind the back line of the field. This requires a combination of a large angle objective (8 mm focal distance) with a big size CMOS imager (1 size).
- Given the size of the shuttlecock (6 centimeters), the distance from the camera to the back-line of the field and the target positioning accuracy determines the required camera resolution.
- The frame rate is determined by the shuttlecock speed and average shuttlecock travel time per trajectory.

To be able to predict the point where the shuttlecock trajectory will intersect the robot range at least five 3-D points are needed. The determination of these points will introduce an initial delay which has to be kept to a minimum.

Given the previous considerations, the optical system has been built using two black and white, 1.4 Mpixels, 100 Hz frame rate, Camera-Link CMOS digital cameras from Photon-Focus. The cameras are hanging at 4m high, against the ceiling, behind the human player (see Fig. 1).

To reduce the number of false positives the environment has also been conditioned. Most of the background has been made black or of dark color. Typically this reduces the number of objects remaining after the filtering process to between one and two.

A test was performed also with colored shuttle and color cameras to reduce even further the number of false positives. However the option to use color cameras was discarded due to the significant price difference between the color and black-and-white cameras at the required resolution and sampling rate. It was also found that the trajectory estimation algorithm described later in section 4 is very successful at eliminating such false positives based on their dynamics.

The images are acquired and processed on a National Instruments real-time platform. The image processing software has been developed in Lab-View. The BLOB analysis algorithm is built as a pipeline consisting of several filtering stages.

(1) The first stage finds all moving objects in the image. The following stages will filter the found objects on size, shape and intensity. The analysis is first performed independently on the images coming from the 2 cameras. An epiline is computed for every object found in the image.
(2) During the second step the information from the images of the 2 cameras is combined. The objects matched in pairs based on the distance between the epilines.
(3) The epilines are computed knowing the relative position between the 2 cameras. The position is expressed as a translation and a rotation matrix. These matrices are computed by performing extrinsic camera calibration. This process is performed by localizing the position of objects with known coordinates in the images taken with the 2 cameras. A 3-D Cartesian model is then fitted between these points where the fit parameters are the matrix coefficients. The fit error is in average below 2cm in all 3 dimensions.

The coordinates are always paired with a time-stamp. The time-stamp corresponds to the moment when the cameras are triggered. The two cameras are triggered simultaneously using an external trigger signal.

The suspected object 3-D coordinates and the corresponding time-stamp are then passed to the shuttlecock trajectory estimation algorithm.

4. TRAJECTORY ESTIMATION

Using the shuttle locations and time-stamps received from the visual system, the trajectory estimation block estimates the position of the shuttlecock at the last received time-stamp and predicts its trajectory.

Estimation and prediction of the trajectory are both based on the differential equations for shuttlecock motion derived in Cooke (2002). The equations in Cooke (2002) describe 2D motion and include the aerodynamic effects of lift, drag and pitch. A comparison between measurements of shuttlecock trajectories obtained with the localization system and simulations with the 2D equations of Cooke (2002) showed that the effect of lift can be neglected. On the other hand, the comparison also showed that shuttlecock motion is not always 2D. All shuttlecocks are designed to spin clockwise, and therefore experience a sideways drift. For certain shots, this sideways drift is not negligible. Hence, we decided to extend the 2D equations of Cooke (2002) to 3D and introduced terms that take the sideways drift due to the shuttlecock spin into account.

A shuttlecock played by the human opponent enters the range of the robot approximately one second after the hit. Due to the strict acceleration and speed limitations, the robot should start its movement along the linear axis as early as possible. Hence, the direction in which the robot must move along this axis should be known as soon as possible after the hit. Therefore, the estimation of the shuttlecock trajectory is based on a estimation algorithm with fast convergence: the Kalman filter. Since the equations of shuttlecock motion are nonlinear, we used a nonlinear extension of the Kalman filter, i.e. the extended Kalman filter (EKF).

The estimation algorithm must continuously track the motion of the shuttlecock, i.e. it should detect hits by the human opponent or by the robot and should continue tracking the shuttlecock after the hit. In the same time, the algorithm must be able to detect false-positive measurements obtained from the localization system. Note that making the distinction between a false-positive measurement and a sudden change in position due to a hit is not always straightforward. The approach that was followed in order to make this distinction, consists in using not a single EKF, but multiple EKFs in parallel. The idea is that one EKF is tracking the actual shuttlecock trajectory while other EKFs are started up on-the-fly (depending on the data received by the localization system) in order to explore the possibility that a hit has occurred.

Note that the use of time-stamps at which the cameras are triggered together with the prediction of the shuttlecock trajectory yields an easy and effective way to deal with delays due to the localization system. In fact, the delays due to the localization system need not to be known, they are accounted for implicitly by the use of the time-stamps and the prediction of the trajectory.

Figures 3 and 4 illustrate the effectiveness of the estimation algorithm. In these figures, the shuttlecock is first flying towards the human opponent (from left to right) and is then hit back by the human at an x coordinate of around 9m. The blue bullets denote the measurements received from the localization system (note the two false-positive measurements on the right). The green stars denote the shuttlecock positions estimated by the algorithm. The red line shows the predicted trajectory starting from the last estimate (note that both figures contain measurements, i.e. blue bullets, that are not yet assumed to be available at that time instant, yet they are shown in order to evaluate the accuracy of the predicted positions). The top figure shows the predicted trajectory just after the hit by the human has been detected. This prediction is clearly off from the true trajectory, but is yet accurately enough to start moving the robot along the linear axis. The bottom figure shows the predicted trajectory 200ms after the hit has been detected. Clearly, the predicted trajectory has converged.

5. INTERCEPTION SYSTEM

The task of the interception system is to determine the hit time and the robot configuration (i.e. hit angle, rotational angle and coordinate along the linear axis) in order to hit back the shuttle to the human opponent at a point along the predicted trajectory. The interception system internally makes use of a model that converts points in world coordinates to the corresponding robot configuration. Based on the mechanical design and the degrees of freedom, it can be easily seen that a point in world coordinates corresponds to at most two robot configurations. The approach used to determine the optimal interception point consists in considering a fixed number of points on the intersection between the robot range and the predicted

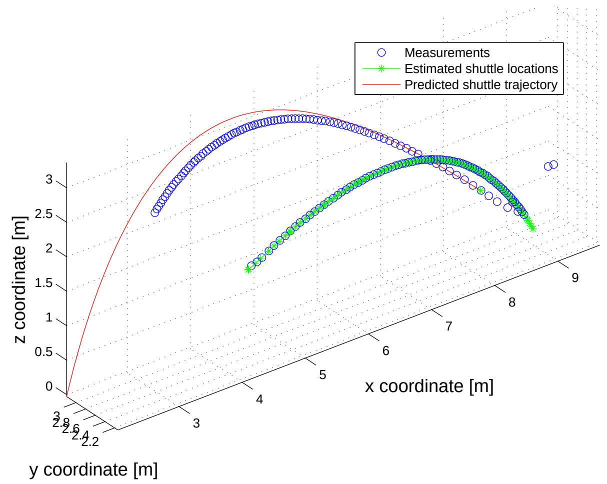

Fig. 3. Predicted shuttle trajectory just at the moment that the hit by the human opponent is detected.

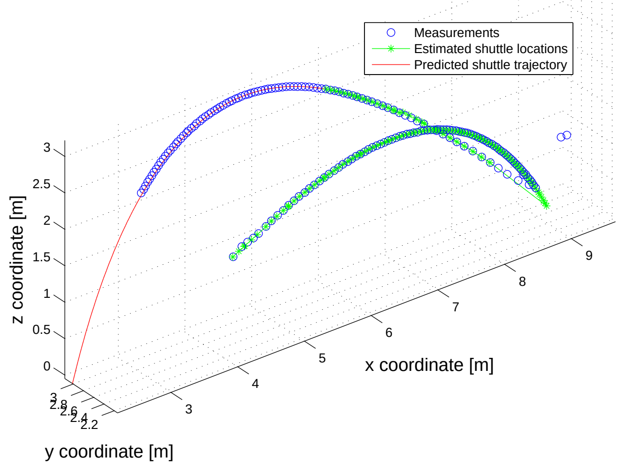

Fig. 4. Predicted trajectory 200ms after the hit has been detected.

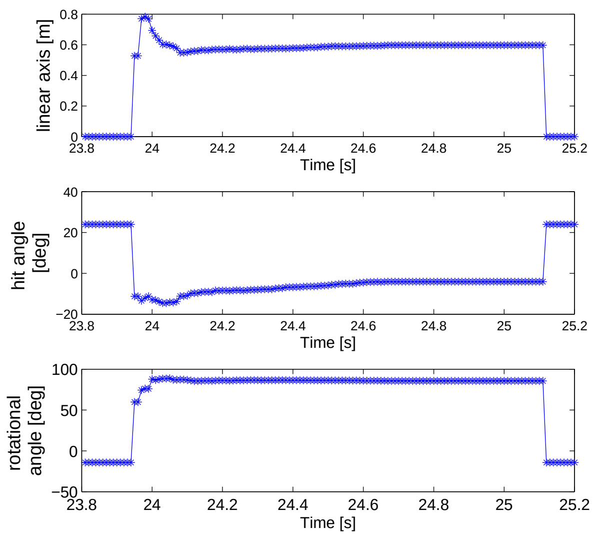

Fig. 5. Simulation of the time evolution of the robot configurations determined by the interception system.

trajectory and to select the point that scores best on the following criteria (in order of importance):

- The robot should be able to reach the interception point in time, i.e. the robot must be able to arrive at the interception point before the shuttlecock flies through the point. In order to check this criterion, an analytical formula for the robot travel time was derived.
- The configuration of the robot must be such that the angle between the velocity vector of middle of the racket and a desired velocity vector is minimized.

Figure 5 shows a simulation of the time evolution of the robot configurations determined by the interception system. The last 0.5s before the hit, the angles typically change by no more than 5° and the coordinate along the linear axis by no more than 5cm.

6. ACTUATION SYSTEM

The actuation system receives the target coordinates of the interception point and positions the robot at the location where the hit motion should happen.

The main control challenges are twofold:

- the total time for the move is limited by the expected fly-time of the badminton shuttle, which is assumed to be 1 second on the average;
- the high degree of uncertainty especially at the beginning of the move when the shuttle trajectory is not well known, as shown of Figures 3 and 4.

Thus the motion trajectory should be time-optimal, but in the same time it should be adjusted *on the fly*, as the motion occurs, by accommodating the latest information propagated trough the visual, trajectory estimation and interception sub-systems. The motion controller should be able to take into account and compensate disturbances that are due to (possibly nonlinear) coupling between axes and the approximate modeling of mechanical parts. In the same time it is required to be sufficiently simple to be implementable in real time.

As the current mechanical design of the robot was performed in such way, so that there is minimal coupling between the dynamics of the different axis, then motion controllers can be designed separately, taking into account the uncertainty due to the small remaining coupling.

The basic problem is to design a fast time optimal control under state and input constraints. Such controllers are inherently non-linear and special care has to be taken for their stability and safety properties. Several possible approaches are tested for the robot.

6.1 System Identification

Figures 6, 7, 8 show the experimental frequency response measurements of robot axes. Several remarks are due on these plots. All of them are significantly different from the classical *double-integrator system*, which is often used to represent such mechanical constructions for control design. From the velocity and timing requirements, the open-loop controller bandwidth was estimated in the range of 50 to 100 Hz.

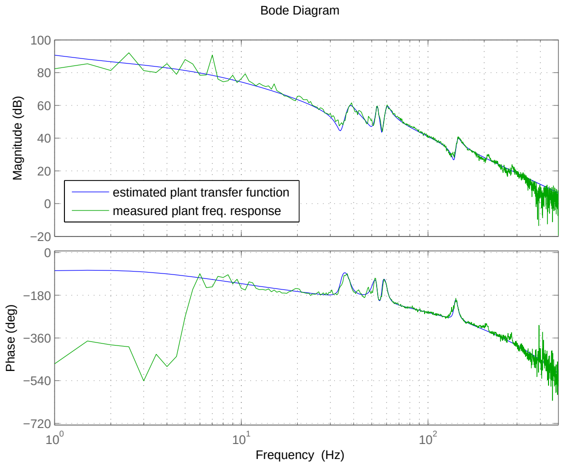

Fig. 6. Linear axis experimental FRF from motor current to linear position

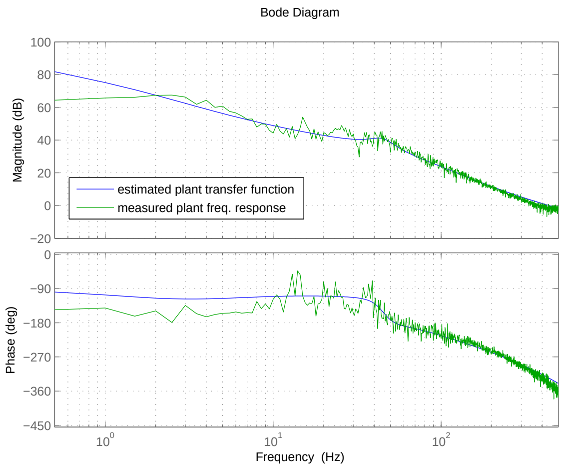

Fig. 7. Rotation axis experimental FRF from motor current to angular position

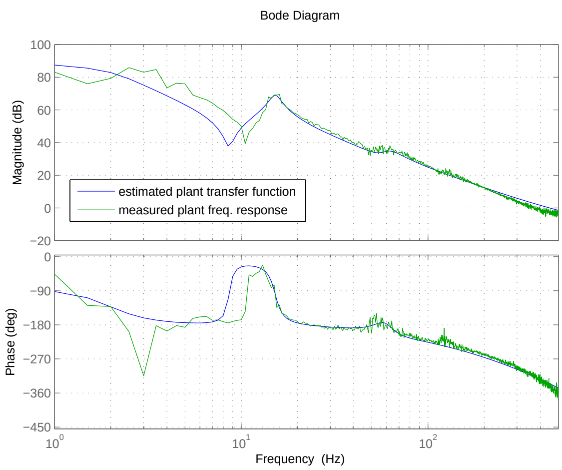

Fig. 8. Hit axis experimental FRF from motor current to angular position

6.2 Optimal Control in the feedback loop

Different optimal control strategies are known and can be applied to the robot feedback loop. Since a basic requirement for the robot is the capability to intercept the shuttle in one second, time-optimal and related strategies will be most important ones.

Multi-parametric piecewise-linear controller (MPT) The MPT approach due to Kvasnica et al. (2004) offers several attractive properties:

- *Flexible* – it offers a range of performance objectives (linear, quadratic, minimum time), uncertainty models, constraints;
- *Sound theory* – it is based on extensible general theory for large class of systems dynamics;
- *Implementable* – the resulting control laws can be embedded in the general form of a C code, or deployed to target platforms using Real Time Workshop;

The weak points for MPT are due to the complex properties of the obtained control laws:

- *Computational complexity grows with high sampling rate* – the optimality of the control laws is strongly related to the look-ahead time of the controller. With fast sampling rates, the number of steps required to look ahead grows very fast and corresponding control law complexity follows. The result may become unimplementable given the restrictions of the real-time implementation.
- *Computational complexity due to high system dimensionality* – the resulting control law complexity grows very fast with the dynamic order of the plant (state-space dimensions).
- *Stability restrictions* – the control laws assume full state feedback. However taking into account the exact stability properties of the hybrid systems including observers is not trivial.

Proximate Time-Optimal Servomechanism (PTOS) The (PTOS) approach from Workman (1987) is another option. The advantages of PTOS are as follows:

- *Maturity* – it is well tested approach with numerous industrial applications ranging from hard disk drives to robotics;
- *Simplicity* – the resulting controllers have been implemented both in continuous time and in discrete time on very limited hardware;
- *Robustness* – the method was extended to handle different classes of systems and was proved to be extremely robust;

The disadvantages of PTOS are related to its inherent relationship with classical double integrator systems. The method was extended to higher order systems, but limitations remain.

Other approaches are also under consideration for the future versions of the robot, such as Model Predictive Control, Sliding Mode Control.

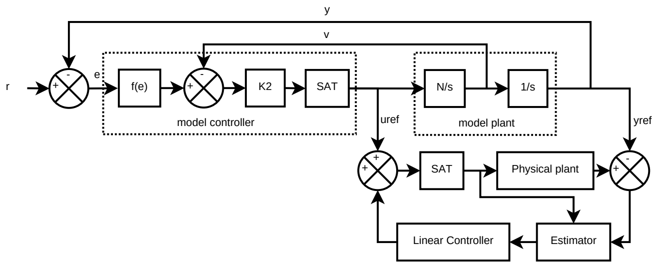

Fig. 9. Model Reference Control System from Dhanda and Franklin (2009)

6.3 Optimal Control out of the feedback loop

The plots on Figures 6, 7, 8 indicate that the plant can not be easily approximated by a double integrator system, as required by the classical PTOS approach above. In the same time, the dynamic order of the systems transfer functions on Figures 6, 7, 8 is in the range of 8, which would lead to very high complexity of the MPT controllers described above.

Therefore a safer and more practical approach is to generate simplified time-optimal reference trajectories out of the feedback loop and use linear feedback controller. The advantage of such approach is that the feedback loop is linear and thus one can expect less surprises from stability point of view, while keeping the complexity of the time-optimal generating part relatively low. The linear loop can be designed using for example H^2, or H^∞ theory, as well as more classical PID control.

For the reference trajectory generation many possibilities are available including splines, polynomial and sine-cosine profiles. However not all of these can accommodate the continuously adjusted information from the trajectory estimation and interception system.

In practice the time-optimal feedback approaches mentioned above in Section 6.2 can be used for trajectory generation, assuming a simplified double-integrator plant. Such a scheme is shown on Figure 9.

In all cases care should be taken to avoid integrator windup and control saturation in the linear loop. Practical ways to achieve this and discussion can be found in Astrom and Wittenmark (1996).

The actual robot first version is currently using two control approaches in this group of methods, which will be described briefly below.

Model Reference Control System (MRCS) This is approach shown on Fig. 9 for the simplified case when PTOS is used for reference generator and the assumption is that the system can be simplified to double integrator. The logic is the same when the system is replaced by a more complex one and/or the controller has more complex structure, for example MPT-like. The core idea in this case is that not only reference trajectories are generated, but also feed-forward signals. This improves the response time of the system, but care should be taken on how well the physical plant is represented by the model plant.

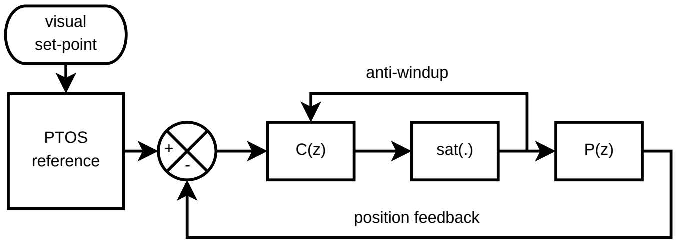

Fig. 10. PTOS reference trajectory generation

On-line Trajectory Generation and tracking The simplified classical version of MRCS is to use only the position y (and possibly velocity v) signals on Fig. 10. It can be regarded as a slightly less-performing variant of MRCS above. Both PTOS and MPT can be used to generate the trajectory assuming ideal double integrator for the plant dynamics.

7. SOFTWARE

There are several practical challenges in the implementation of the system, which are also to be taken into account – control software and hardware and the communication between different components.

The diversity of sub-systems is reflected in the soft-wares used. It is very challenging to find a single vendor, who can provide end-to-end solutions for all the requirements set forward by the different tasks, thus the distributed multi-platform approach was adopted. In addition to that, the computational load is quite significant.

Special effort was made to keep the costs low, while extracting maximum performance from the hardware. The motion control algorithm was tested and works using two alternative software platforms by IGH-Essen (2006) and Soetens (2006). Both of these run on a real-time Linux system (RTAI) with a dual-core Intel-based PC. Attempt was made to keep at least the software platform from a single vendor and tests were performed also using National Instruments CompactRIO platform for the motion controller. The results had limited success due to the limited computing power of the CompactRIO hardware.

The communication between the the visual sub-system, running on a dedicated computing quad-core Intel-based platform under Real-Time LabView, is done using UDP packets on a dedicated Ethernet line at 100 Hz sampling rate. Such a configuration has very low delay and high determinism.

The communication between the controller PC and the robot motor hardware is performed using Ethercat technology at 1000 Hz sampling rate.

8. CONCLUSION

We have presented the first version of a robot playing badminton. The design of such a system contains diverse problems - mechanical, visual, estimation and strategy, motion control, software and is a challenging case in Mechatronics design both from theoretical and from practical perspective. Future version of the robot are currently under development, with even more challenging goals, such as wireless information transfer, completely autonomous platform and efficient energy recuperation.

REFERENCES

Astrom, K.J. and Wittenmark, B. (1996). *Computer Controlled Systems*. Longman Higher Education.

Chesi, G. and Hashimoto, K. (eds.) (2010). *Visual Servoing via Advanced Numerical Methods*. Number 401 in Lecture Notes in Control and Information Sciences. Springer.

Cooke, A. (2002). Computer simulation of shuttlecock trajectories. *Sports Engineering*, 5, 93–105.

Dhanda, A. and Franklin, G.F. (2009). An improved 2–dof proximate time optimal servomechanism. *IEEE Transactions on magnetics*, 45(5), 2151–2164.

IGH-Essen (2006). Etherlab. `http://www.etherlab.org`.

Kvasnica, M., Grieder, P., and Baotić, M. (2004). Multi-Parametric Toolbox (MPT). `http://control.ee.ethz.ch/~mpt/`.

Soetens, P. (2006). RTT: Real-Time Toolkit. `http://www.orocos.org/rtt`.

Workman, M.L. (1987). *Adaptive Proximate Time-Optimal Servomechanisms*. Ph.D. thesis, Stanford Univ., Stanford, CA.