Comparison of Stochastic Methods for Control in Air Traffic Management [⋆]

Alison Eele [*] **Jan Maciejowski** [*]

[*] *Department of Engineering, University of Cambridge, Cambridge, UK (e-mail: {aje46,jmm1}@cam.ac.uk).*

Abstract: This paper provides a direct comparison of two stochastic optimisation techniques (Markov Chain Monte Carlo and Sequential Monte Carlo) when applied to the problem of conflict resolution and aircraft trajectory control in air traffic management. The two methods are then also compared to another existing technique of Mixed-Integer Linear Programming which is also popular in distributed control.

Keywords: Air Traffic Control, Stochastic Control, Optimal Trajectory, Model Predictive Control, Aerospace Trajectories.

1. INTRODUCTION

Air Traffic Management (ATM) is concerned with the routing, safety and scheduling of aircraft in regions of airspace. Currently this role is performed by Air Traffic Control (ATC) and is a very human orientated process though steps towards autonomy have been made in recent years Atkin et al. (2008). ATC has complete control of the aircraft through all stages of flight, from pre-flight flight plans, updates and additional instructions based on traffic flow to landing scheduling. The majority of ATC concern lies with the avoidance of dangerous encounters through maintenance of safe separation between aircraft. Loss of the minimum safe separation between two or more aircraft is defined as a conflict. If a conflict does not have safety repercussions (for example in mid-long term detection) the priorities of the aircraft are considered by the ATC when resolving the conflict. These priorities could include avoidance of turbulence or desired time of arrival.

It is acknowledged that the current system of ATM is near the upper limit of traffic it can safely accommodate (European Commission (2000)). The level of anticipated growth in aviation travel is debated, however it is currently agreed that it will increase in the next 20 years by up to 200% (Federal Aviation Authority (2009)). It is untenable to accommodate this anticipated traffic growth without a shift in paradigm. The aviation authorities have also set aggressive targets to handle the traffic growth whilst achieving cuts in emissions (Advisory Council for Aeronautics Research in Europe (2002)). A portion of these emission cuts will come from the next generation of ATM.

The role of ATC is further complicated by the introduction of uncertainty. This uncertainty is introduced by the effect of the wind, incomplete knowledge of the physical coefficients of the aircraft and imprecision in the execution of ATC commands. With the addition of uncertainty tasks such as conflict detection become a matter of assessing the probability of a conflict given the current state of the airspace and the uncertainty of the future positions of the aircraft. Probabilistic models can be used to estimate the distribution of the future state of the aircraft given the current state.

In model predictive control (MPC) an open loop optimal control problem is solved at each time step (Maciejowski (2002)). This optimisation problem can be nonconvex, with nonlinear dynamics, constraints and an objective function. Such optimisation problems are challenging to solve and can require sophisticated optimisation techniques. This paper is concerned with using stochastic optimisation methods to solve the control problem at each time step. Stochastic optimisation allows for modelling the uncertainty present in ATC whilst also able to cope with nonlinear dynamics, constraints and an objective function.

Two forms of Stochastic Optimisation have previously been proposed for use in an MPC framework. The first, Markov Chain Monte Carlo (MCMC), is a set of methods for sampling from probability distributions based on constructing a Markov Chain that has the desired distribution as its equilibrium distribution. The state of this generated chain after a large number of iterations is then sampled to yield a nearly-optimal action (Müller et al. (2004)). MCMC has been used on ATC problems previously by Lecchini-Visintini et al. (2006), where it demonstrated capability for distinguishing between different manoeuvres and their associated costs through density of samples. Other work has focused on MPC stability analysis when MCMC is used Siva et al. (2009) with demonstration on another trajectory planning application. Current research into MCMC focuses predominantly on attempting to find finite-time guarantees for convergence (Lecchini-Visintini et al. (2007, 2010)).

Sequential Monte Carlo (SMC) optimisation is better known for its use in 'particle filtering' for model estimation (Doucet et al. (2000) and Kantas et al. (2009)). It was observed that MPC optimisation shares similar features to that of model estimation and through this connection

[⋆] Work supported by EPSRC (Engineering and Physical Sciences Research Council - UK) Grant No. EP/G066477/1

SMC was applied to MPC optimisation by Kantas et al. (2008). The work of Kantas et al. (2008) included examples of SMC applied to MPC of multiple vehicles flying in two dimensions under the predicted effects of wind and additional Gaussian noise.

Both of the stochastic methods compared are better suited to situations where quick control responses are not required. However there is potential for improving the solve times of both of the MCMC and SMC methods through parallelisation. This paper provides a direct comparison of the two mentioned stochastic optimisation techniques when applied to the problem of conflict resolution in air traffic management. The paper focuses on how to tailor the two methods to the application and comparison of the resulting runtimes.

The paper is organised as follows. Section 2 describes the problem statement. Section 3 reviews both stochastic optimisation methods and introduces the customisation to fit them to the considered problem. Section 4 presents the results of the comparison simulations using the firstly the two stochastic methods and then comparison to another existing non-stochastic method. Finally, Section 5 presents conclusions.

2. PROBLEM STATEMENT

For the sake of comparison a common dynamics model is used between both stochastic optimisation methods. This section reviews the dynamics model used. The aircraft are modelled in discrete time in three dimensions with a constant speed in the horizontal plane. For simplicity the vertical and horizontal dynamics have been decoupled. This is not a limitation of either of the stochastic optimisation methods which can readily accept any nonlinear and non-convex dynamics model as it requires no assumptions on the dynamics or distributions.

The horizontal motion of each of the aircraft is modelled as:

$$X_{k+1} = X_k + VT_s[\sin(\theta_{k+1}), \cos(\theta_{k+1})]^T + W_{k+1} \quad (1)$$

Where X_k is the state, θ_k is the bearing of the vehicle and W_k is the noise at discrete time step k. V is the speed and T_s is the length of the discrete time step. Similarly the vertical motion of the aircraft is described by:

$$H_{k+1} = H_k + h_{k+1} + M_{h,k+1} \quad (2)$$

Where H_k is the altitude at time step k, h_k is the change in altitude and $M_{h,k}$ is the altitude noise. $M_{h,k}$ is mutually independent of W_k. In this paper our attention is restricted to the fully observed case with finite horizon problems for the MPC formulation.

The control inputs are the change in heading, $\delta\theta$, and change in altitude, h. These control inputs are both bounded:

$$\theta^-_{\max} \leq \delta\theta \leq \theta^+_{\max} \quad (3)$$

$$h^-_{\max} \leq h \leq h^+_{\max} \quad (4)$$

Further constraints are imposed upon each aircraft that they must avoid all other aircraft in the problem. The protection zone around each aircraft is modelled as a cylinder throughout this paper with horizontal radius P_r and altitude separation of P_h.

$$\begin{aligned}(x_{i,k} - x_{j,k})^2 + (y_{i,k} - y_{j,k})^2 &\geq (2P_r)^2 \\ \vee |h_{i,k} - h_{j,k}| &\geq 2P_h \\ \forall k, \forall i, j \in \{1, ..., N_v\} &: i \neq j \quad (5)\end{aligned}$$

Each aircraft is given an initial condition:

$$X_{0,i} = [x_{0,i}, y_{0,i}, h_{0,i}]^T \forall i \in \{1, ..., N_v\} \quad (6)$$

and an aircraft is deemed to have finished its path once it has reached a terminal set if there exists a k such that:

$$\begin{aligned}(x_{T_f,i} - x_{k,i})^2 + (y_{T_f,i} - y_{k,i})^2 &\leq G_r^2 \\ |h_{T_f,i} - h_{k,i}| &\leq G_h \\ \forall i &\in \{1, ..., N_v\} \quad (7)\end{aligned}$$

Finally we determine a non-negative open loop cost. This paper assumes that the aim is to find the shortest distance between the start to the goals for each of the vehicles.

$$J_{\text{dist}}(k) = \sum_{i=0}^{Nv} ||[X_{k,i}H_{k,i}] - [X_{T_f,i}H_{T_f,i}]|| \quad (8)$$

This is then used as a non-negative cost for the MPC formulation with N_s steps in the horizon length by adding the costs for each of the N_s steps then scaling.

$$J_{k:k+N_s} = \sum_{s=k}^{k+N_s} J_{\text{dist}}(s)/(s + 1 - k) \quad (9)$$

The importance of terms are scaled to put greater emphasis on the earlier steps in the horizon with greater concern for minimising the effects of noise. Without this scaling each step in the horizon carried the same weight in the cost. Noise which affects the $\delta\theta$ in the early steps has a large effect on the final position of the vehicle at the end of a horizon. By giving a higher cost to these early steps the cost seeks to mimic simple robust control methods. The weighting of each step decays slowly to avoid later steps having no effect on the cost. The objective of our optimisation is to find the controls $\delta\theta$ and h which minimise $J_{k:k+N_s}$

$$J^{\min}_{k:k+N_s} = \min J_{k:k+N_s} \quad (10)$$

3. STOCHASTIC METHODS

3.1 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a set of methods for sampling from probability distributions based on constructing a Markov Chain that has the desired distribution as its equilibrium distribution. The state of this generated chain after a large number of iterations is then used as a sample from the desired distribution.

Our optimisation problem to minimise cost function J previously defined in Section 2 is unsuitable for MCMC as this requires a maximisation problem. However a simple transformation of variables can change (9) into a maximisation problem.

$$g_{k:k+nS} = -J^{\min}_{k:k+nS} + \sup_{X,H,\delta\theta,h} J_{k:k+nS} \quad (11)$$

Unlike the standard negation of a minimisation to yield a maximisation the MCMC method and SMC method both still require a non-negative cost. To ensure this the maximum possible value of the cost is added. Typically it is difficult to calculate exactly the maximal value of the cost. However due to the nature of the application and the limited horizon we can approximate this simply by measuring the furthest distance from the goals the vehicles can reach in the number of horizon steps ignoring turning and avoidance constraints. Similarly the minimum cost would be the closest distance the vehicles can reach to the goals again disregarding turning and avoidance constraints (used later in (12)).

This cost function however behaves somewhat differently at different stages of the vehicles' travel. Ideally we want the cost function to distinguish well between poor paths from good paths. When the vehicles are far from the goal, the dominating factor is that of the overall distance left to the goal. The effect of noise or poor control decisions are poorly scaled in comparison. As the vehicle gets closer to the goal this effect is gradually negated. To try and offset this the cost function will be normalised and rescaled between the maximum possible distance the vehicle could be from to the goal given its current starting point, and the minimum possible distance it could be to the goal from its starting point.

$$g_{k:k+nS} = \frac{-J^{\min}_{k:k+nS} + \sup_{X,H,\delta\theta,h} J_{k:k+nS}}{\sup_{X,H,\delta\theta,h} J_{k:k+nS} - \inf_{X,H,\delta\theta,h} J_{k:k+nS}} \tag{12}$$

In MCMC there are two main methods of handling constraints which cannot be dealt with by distribution design. The first, resampling, requires each sample to be tested to see if it obeys the constraint, and rejected if it does not. A fresh sample is then drawn from the prior distribution and the process repeats. This method is acceptable in situations where constraints are few and the samples unlikely to be close to constraint boundaries. Alternatively, the second method is that of penalty functions. In this case instead of rejecting outright a sample which violates a constraint it is instead penalised, which in turn affects the chance of it being accepted. The penalty functions do weaken the guarantee that a conflict could not occur between two vehicles. However the penalty can be increased incrementally as the number of iterations and samples increase in order to discourage the propagation of samples where the constraints have been breeched. The vehicle avoidance constraints are ideal candidates for implementation as penalty functions. In the following augmentation of the cost equation (9) a penalty per constraint broken has been added.

$$\hat{J}^{\min}_{k:k+nS} = \min \sum_{s=k}^{k+nS} \big(\frac{(J_{\mathrm{dist}}(s) + N_C * \mathrm{penalty} * j)}{(s+1-k)}\big) \tag{13}$$

Where N_C is the number of constraints broken and j is the current number of samples. As the sample schedule increases the level of penalty associated with breaking a constraint also increases. This alters equation 12 to:

$$g_{k:k+nS} = \frac{-\hat{J}^{\min}_{k:k+nS} + \sup_{X,H,\delta\theta,h} \hat{J}_{k:k+nS}}{\sup_{X,H,\delta\theta,h} \hat{J}_{k:k+nS} - \inf_{X,H,\delta\theta,h} \hat{J}_{k:k+nS}} \tag{14}$$

Algorithm 1 shows the implementation of MCMC into an MPC framework for the problem in question. It incorporates a Metropolis-Hastings update in Steps 20-21, described in greater depth in Robert and Casella (1999).

Algorithm 1. Markov Chain Monte Carlo MPC

```
 1: select a prior distribution $q_0(\cdot)$
 2: $k \leftarrow 0$
 3: Define a monotonically increasing SampleSchedule of length $J_{\max}$
 4: while all vehicles have not reached their terminal sets do
 5:     $k \leftarrow k+1$
 6:     $j \leftarrow$ SampleSchedule(1)
 7:     Draw a sample for $\delta\theta_0$ and $h_0$ (collectively considered as $U_0$) from $q_0$
 8:     $U \leftarrow U_0$
 9:     Generate j copies of the samples and apply j realisations of noise to them
10:     Evaluate the cost of the each of the j realisations
11:     The total cost of the sample $g = \prod_{i=1}^{j} J_{k:k+nS}(i)$
12:     $g \leftarrow g^{(1/j)}$
13:     for n = 1 to maxIterations do
14:         $Ja \leftarrow$ SampleSchedule$(n)$
15:         Draw a new sample $\tilde{U}_n \sim q(\cdot\,|U)$
16:         Generate j copies of the samples and apply j realisations of noise to them
17:         Evaluate the cost of each of the j realisations using (14)
18:         The total cost of the proposed sample $\tilde{g} = \prod_{i=1}^{j} \tilde{J}_{k:k+nS}(i)$
19:         $\tilde{g} \leftarrow \tilde{g}^{(1/j)}$
20:         Evaluate the acceptance probability $\rho_j = \min\Big[1, \frac{\tilde{g}^j q(U|\tilde{U})}{g^j q(\tilde{U}|U)}\Big]$
21:         Accept proposal $\tilde{U}$ with probability $\rho_j$ and set $U = \tilde{U}$. Otherwise leave $U$ unchanged with probability $1-\rho_j$
22:     end for
23:     apply first step of control of current $U$ to the current position with gaussian noise.
24: end while
```

Many prior distributions (sampled during Step 7) are compatible with MCMC and selecting an appropriate distribution is one of the elements of the tailoring of the method to the specific application. In our application we will be sampling from the prior distribution for values of $\delta\theta$ and h. These are both bounded values due to the constraints outlined in (3) and (4). Therefore the prior distribution selected should be bounded accordingly. To improve the speed of convergence to the equilibrium distribution some previous memory of samples would be useful in the prior distribution thus we have chosen to model the prior distribution as a truncated Gaussian distribution with a mean of the previous accepted sample and fixed variance.

3.2 Sequential Monte Carlo

Sequential Monte Carlo (SMC) optimisation is better known for its use in 'particle filtering' for model estimation. It was observed that MPC optimisation shares similar features to that of model estimation and through this connection SMC was applied to MPC optimisation by Kantas et al. (2008). The work of Kantas et al. (2008) included examples of SMC applied to MPC of multiple vehicles flying in two dimensions under the predicted effects of wind and additional Gaussian noise.

The underlying concept of SMC is to approximate a sequence of distributions of interest as a collection of L discrete masses of the variables (more commonly referred to as particles) weighted by a collection of weights to reflect the shape of the distribution. As the distribution to be approximated can vary with time the weights and particles are propagated iteratively using a sequential importance sampling and resampling mechanism. This sampling and resampling mechanism uses the particles of iteration i-1 to obtain new particles at iteration i. In this way a population of particles is iterated upon until a final sample is drawn from the population to act as an estimator of the maximisers. This final sample is sometimes taken as the mean of the values of the particles however in multi modal distributions this would given an inaccurate representation of the distributions global optimisers. Thus it is essential to select the modal value of the particles and this has been followed in this paper's implementation.

Algorithm 2. Sequential Monte Carlo MPC

```
$k \leftarrow 0$
Define a monotonically increasing SampleSchedule of length $J_{\max}$
while all vehicles have not reached their terminal sets do
    $k \leftarrow k+1$
    Extract random samples for $\delta\theta_{k,i}$ and $h_{k,i}$ for each of the L particles.
    weight $\leftarrow 1/L$ for particles 1 to L
    for $j = 1$ to $J_{\max}$ do
        $j_a \leftarrow$ SampleSchedule($j$)
        Sample gaussian noise on $\delta\theta_{k,i}$ and $h_{k,i}$ for each of L particles.
        for $jj = 1$ to $j_a$ do
            for each step in the horizon length do
                Calculate future positions using (1-2) with additional gaussian noise.
                Calculate cost using (12)
            end for
            for each particle $l = 1$ to $L$ do
                if constraints (3-5) are broken by the particle then
                    weight($l$) $\leftarrow 0$
                end if
                weight($l$) $\leftarrow$ weight($l$) $*$ cost
            end for
            Rescale all weights back to probabilities, such that the sum of the particle weights = 1.
        end for
        Resample the particles based on the weights for a new population of L particles.
        weight $\leftarrow 1/L$ for particles 1 to L
    end for
    Calculate next step using mode of all particles' $\delta\theta_k$ and $h_k$
    Set new position as current position.
end while
```

Unlike MCMC, SMC requires far less tuning to work successfully on applications (observed in the work of Kantas et al. (2008)). Therefore some of the complexities encountered by the MCMC implementation, such as careful choice of distributions and penalty functions is not required here. The SMC method however also relies on maximising an optimisation problem, so like with the MCMC cost function demonstrated in the previous section the cost equation (9) is once more converted into a normalised maximisation problem (12).

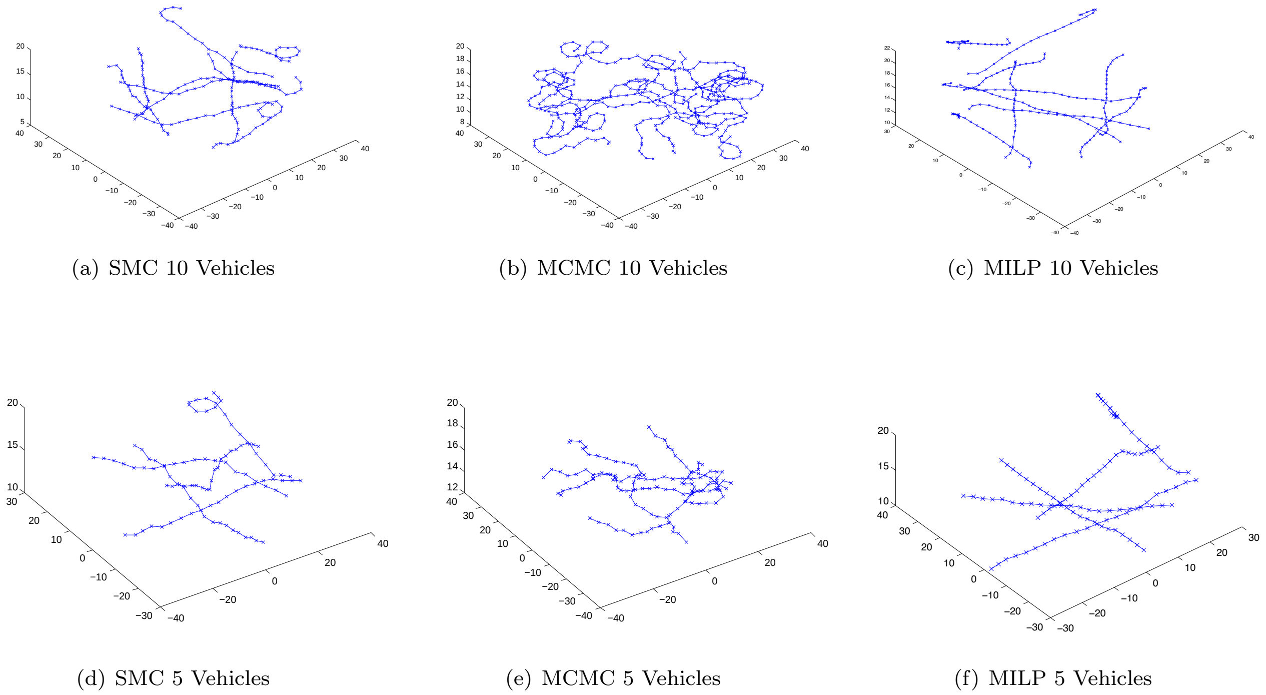
The algorithm used to apply the SMC to the current application is shown in Algorithm 2, where k is the current time step and L is the number of particles.

4. RESULTS

The two stochastic optimisation methods are compared on test cases across a range of vehicle numbers. Both algorithms were given the same test cases to solve. The noise experienced by both methods was drawn from the same distribution however the samples differed between the methods. Each test case had between 3 and 10 vehicles (though some examples involving up to 20 vehicles have been generated). These vehicles were randomly given a starting point and target on the perimeter of a cylinder of height 5 and radius 30. Vehicles were given a starting horizontal heading directly towards the centre of the cylinder and all start and end points were designed such that the vehicles had a minimum distance to travel to encourage the chance of vehicle interaction. As the number of vehicles increased the chance of conflict also increased. No vehicles started off in conflict and no terminal points were in conflict. Both methods were given a horizon length of 6 steps to work with and the vehicles were constrained with the same maximum turning rate and maximum altitude changes. Twenty cases of each individual number of vehicles from 3 to 10 were simulated. The tests were done on a range of computers with 7GB memory and 3GHz speed processors. Both methods were implemented in Matlab.

All cases tested were limited in time to 12 hours (roughly $4.3 * 10^4$ seconds in the figures), beyond this time the problem was stopped and time taken recorded as 12 hours. As previously mentioned and further demonstrated in the results of this work, the runtimes of both stochastic methods do make them unsuited to control applications where fast response times are required. To provide some form of benchmark to existing methods the stochastic optimisation methods were compared to MPC Mixed Integer Linear Programming (MILP). The MILP method used was a receding horizon formulation solved centrally for all vehicles. The horizon used was 20 steps to avoid the need to use a cost-to-go estimate. A linearised version of the model used in Section 2 was implemented using Richards and How (2002) as a basis. The protection zones of the aircraft were approximated as square prisms (although higher polygon prisms could have been used to give a closer approximation to the circular protection region in the horizontal plane). The number of sides used in approximating the cylindrical protection zone of the vehicles directly relates to the number of binaries required to represent the avoidance constraints. A reduced number of binaries improves the runtime of the MILP method for multi vehicle avoidance.

Figure 1 shows example paths generated by the two stochastic methods and MILP on a variety of problem sizes. There are some clear differences between the paths



(a) SMC 10 Vehicles

(b) MCMC 10 Vehicles

(c) MILP 10 Vehicles

(d) SMC 5 Vehicles

(e) MCMC 5 Vehicles

(f) MILP 5 Vehicles

Fig. 1. Comparison of Paths Generated between the Two Stochastic Methods and MILP MPC

generated by MCMC and SMC. The noise rejection of SMC seems to have been better than that of MCMC with most paths direct towards their goals. MCMC paths tended to meander towards their goals, though this could have been due to MCMC not being provided enough time to completely converge on a global maximiser. The behaviour of beginning to circle around the terminal set whilst waiting for other vehicles to reach their terminal sets can be observed in both cases. The effect is greater the larger number of vehicles in the problem. Figure 2 exhibits the loitering characteristic and also demonstrates the number of vehicles that SMC was able to handle in the amount of time allowed. MCMC was unable to solve the 20 vehicle case within the time limit.

Figure 3 shows the runtime comparison between SMC, MCMC and MILP split between the number of vehicles present in each model. It is worth noting only the MCMC method experienced a cut off from the upper limit on solve time. The lowest number of vehicles in a problem which failed to finish when solved by MCMC was 6 vehicles. None of the 9 vehicle cases were solved by MCMC within the time span, however two of the 10 vehicle cases did solve.

The level of variation in time for problem completion increases as the number of vehicles increase, this could be due to a number of factors. As the number of vehicles increase the chance of conflicts also increases and the complication of conflicts would have a direct bearing on the solve times of each of the methods. The runtime variation for SMC remains the lowest throughout the test cases. This is likely due to the minimal change in the amount of computation required in the presence of conflicts.

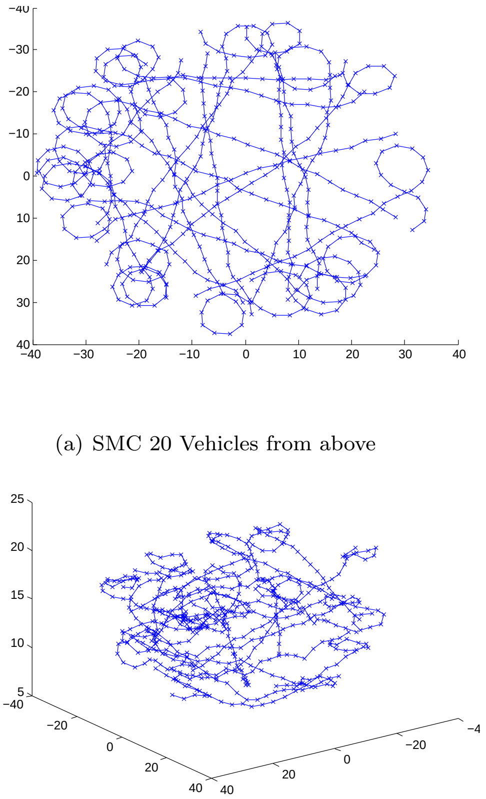
The SMC method performs comparably with the MILP approach and indeed shows a marked improvement on scalability as the number of vehicles increases. Neither MCMC or MILP were able to handle any of the proposed 20 vehicle test cases whereas SMC managed to solve all but one within the allowed time. Even with the improvements of penalty functions and well chosen prior distribution to reduce resampling, MCMC was significantly slower than both the other methods.

5. CONCLUSIONS

In this paper we compared two stochastic optimisation methods, Markov Chain Monte Carlo (MCMC) and Sequential Monte Carlo (SMC), on a variety of air traffic management (ATM) problems with multiple vehicles within a model predictive control (MPC) setup. Both methods were then compared to a MPC mixed-integer linear programming approach. A series of test cases with varying numbers of vehicles between 3 to 10 were used to compare each of the algorithms though SMC was shown to be capable of handling problems with up to twenty vehicles.

Of the two stochastic methods SMC required less tuning to perform well on the proposed ATM application which confirms the statement in Kantas et al. (2008). Of the three methods SMC and MILP achieved the best runtimes. The variance of runtime was also the least in SMC. Solution quality varied quite noticeably between the two stochastic methods with MCMC failing to reject the noise as well as the SMC despite the long time allowed for convergence.

As already observed, methods used in the paper are unsuited for control applications which require a fast response. However all three of the methods compared in this paper can have their runtime reduced. In the case of the stochastic methods future work can focus on advantages of the methods being parallelised. Meanwhile



(a) SMC 20 Vehicles from above

(b) SMC 20 Vehicles from the side

Fig. 2. Twenty vehicle problem solved by SMC viewed from two different angles

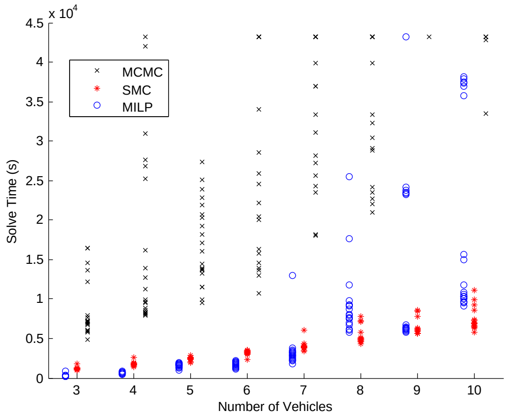


Fig. 3. Comparison of Solve Time Between Three Methods

the MILP method benefits greatly in runtime through distributed control between each of the vehicles Kuwata et al. (2007)

6. ACKNOWLEDGEMENTS

The authors would like to acknowledge Nikolas Kantas and Ellini Siva for their assistance with the stochastic optimisation methods in this paper.

REFERENCES

Advisory Council for Aeronautics Research in Europe (2002). *Strategic Research Agenda 1*, volume 2, chapter The Challenge of Air Transport System Efficiency. (ACARE).

Atkin, J.A.D., Burke, E.K., Greenwood, J.S., and Reeson, D. (2008). On-line Decision Support for Take-Off Runway Scheduling with Uncertain Taxi Times at London Heathrow Airport. *Journal of Scheduling*, 11(5), 323–346.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. *Statistics and Computing*, 10, 197–208.

European Commission (2000). SESAR. Office for Official Publications of the European Communities, Luxembourg. ISBN 92-894-0376-4.

Federal Aviation Authority (2009). FAA Aerospace Forecasts for years 2009-2025. Technical report, Federal Aviation Authority.

Kantas, N., Doucet, A., Singh, S., and Maciejowski, J. (2009). An Overview of Sequential Monte Carlo Methods for Parameter Estimation in General State-Space Models. In *In Proceedings of IFAC Symposium on System Identification*. Saint-Malo, France.

Kantas, N., Maciejowski, J., and Lecchini-Visintini, A. (2008). Sequential Monte Carlo for Model Predictive Control. In *International Workshop on Assessment and Future Directions of NMPC*. Pavia, Italy.

Kuwata, Y., Richards, A., Schouwenaars, T., and How, J. (2007). Distributed Robust Receding Horizon control for Multi-Vehicle Guidance. *IEEE Transactions on Control Systems Technology*, 15(4).

Lecchini-Visintini, A., Glover, W., Lygeros, J., and Maciejowski, J. (2006). *Monte Carlo Optimisation for Conflict Resolution in Air Traffic Control*, 257–276. Springer Berlin/ Heidelberg.

Lecchini-Visintini, A., Lygeros, J., and Maciejowski, J. (2007). Simulated Annealing: Rigorous Finite-Time Guarantees for Optimization on Continuous Domains. In *Advances in Neural Information Processing Systems, Proceedings of NIPS*, 20.

Lecchini-Visintini, A., Lygeros, J., and Maciejowski, J. (2010). Stochastic Optimization on Continuous Domains with Finite-Time Guarantees by Markov Chain Monte Carlo Methods. *IEEE Transations on Automatic Control*.

Maciejowski, J. (2002). *Predictive Control with Constraints*. Pearson, Prentice Hall.

Müller, P., Sansó, B., and Iorio, M.D. (2004). Optimal Bayesian Design by Inhomogeneous Markov Chain Simulation. *Journal of the American Statistical Association*, 99(467), 788–798.

Richards, A. and How, J. (2002). Aircraft Trajectory Planning with Collision Avoidance using Mixed Integer Linear Programming. In *Proceedings of the American Control Conference*, volume 3, 1936 – 1941. Anchorage, Alaska. doi:10.1109/ACC.2002.1023918.

Robert, C. and Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer.

Siva, E., Goulart, P., Maciejowski, J., and Kantas, N. (2009). Stability of Model Predictive Control using Markov Chain Monte Carlo Optimisation. In *in Proceedings of the European Control Conference*. Budapest.