On the Construction of Discrete-Time ARMA Models from Continuous-Time State-Space Models

Ralph J. McDougall
Department of Engineering
University of Cambridge
Cambridge, UK
rjm279@cam.ac.uk

Simon J. Godsill
Department of Engineering
University of Cambridge
Cambridge, UK
sjg30@cam.ac.uk

***Abstract*—Data processing systems typically perform inference on signals using discrete-time stochastic models. This provides a challenge when the same real-world process is modelled differently by separate observers, as can be the case in multi-sensor systems. Stationary ARMA processes are well understood, with known algorithms for converting them to a particular class of continuous-time process with identical covariance structure. These ideas can be extended to provide ARMA models for integrated processes, as used, for example, in object tracking applications, and financial modelling. The sequence of successive predictive distributions of the process is of particular interest in tracking applications, and we show how to find equivalent state-space representations for ARMA (discrete-time) and CARMA (continuous-time) processes by matching these predictive distributions. Generic numerical algorithms for converting between ARMA and CARMA models are presented, as well as conversion of the latent process states between these discrete and continuous-time systems. These are shown to be consistent with the known algorithms for stationary processes and can also be applied to non-stationary processes.**

***Index Terms*—stochastic processes, ARMA, CARMA, SDE, SSM, object tracking, numerical methods**

I. Introduction

Observed data signals vary continuously with time, with modern data processing systems sampling them at (typically) regular time intervals to produce discrete-time observation processes. The model underlying the observed stochastic process is then either learned or assumed known *a priori* when the process is observed in noise. This raises practical challenges when the same signal is observed by sensors with different underlying models, owing, for example, to differing sampling rates. It is natural to model the underlying continuous-time processes as stochastic differential equations (SDEs) [1], with the discrete-time sampled process model acting as a proxy for the SDE. Fitting SDE parameters from data, in the presence of noise, is a hard problem with applications in finance, econometrics, and physical systems [2]–[5]. It may be simpler to consider the observed process as inherently being a discrete-time process, for which a common class of models is the autoregressive moving average (ARMA) process. There is a vast literature on fitting ARMA parameters, and in particular, autoregressive (AR) parameters [6], [7].

A natural question arises as to which inferred ARMA models are exactly equivalent to linear SDEs in some sense. It has long been known that a class of linear SDEs, appropriately known as continuous ARMA (CARMA) models, can always be sampled to produce ARMA models with identical temporal covariance structure when they are stationary [8]. These models will be described in Section II. The inverse problem, i.e. which ARMA models are linear SDEs in the temporal covariance sense, remained open for decades until an explicit algorithm for model conversion was found [9]–[11]. Integrated processes have seen applications in finance [12], and are of particular interest as these provide natural descriptions for object positions [13], [14], however their non-stationarity makes them difficult to analyse using the standard tools. These ideas are briefly discussed in Section III.

ARMA models, and in particular the AR model, have seen extensive use as a basis for tracking and prediction [15]–[18], with inference being typically performed by the Kalman Filter [7]. Switching between different (equivalent) models of the process also requires converting the inferred state between the different models. The prior work on finding equivalent ARMA and CARMA models typically does not discuss the transformation between equivalent states. Of particular interest in tracking and decision-making systems is the predictive performance of the underlying models. This work provides numerical algorithms for determining models that give numerically identical short-term predictive distributions for the observed process, as well as the equivalent state distributions needed to obtain those predictive distributions in Section IV. Our algorithm is shown to be consistent with that of [9] for stationary processes, and it can be used to find the parameters for a ARMA representation of common path models from object tracking in Section V.

II. Stochastic Time Series Models

A signal $x : \mathbb{R}^+ \to \mathbb{R}$ is generated by a CARMA(p, q) process, X_t, if it satisfies the following relation in terms of the differential operator, D:

$$a_c(D)X_t = b_c(D)W_t, \tag{1}$$

$$a_c(s) \triangleq s^p - \sum_{i=0}^{p-1} a_i s^i, \quad b_c(s) \triangleq s^q + \sum_{i=0}^{q-1} b_i s^i. \tag{2}$$

where W_t is a white-noise process with variance σ_c^2. Indeed, W_t can be a white Lévy process with finite second moment [19], [20], although the focus here is on Brownian noise. The process X_t can also be written as the output of a state-space model (SSM) in terms of a latent p-dimensional vector process, $\boldsymbol{Z}_t$:

$$d\boldsymbol{Z}_t = \boldsymbol{A}_c\boldsymbol{Z}_t dt + \boldsymbol{e}dW_t, \tag{3}$$
$$X_t = \boldsymbol{b}_c\boldsymbol{Z}_t, \tag{4}$$

with the system matrices of the form:

$$\boldsymbol{A}_c = \begin{bmatrix} a_{p-1} & a_{p-2} & \cdots & a_1 & a_0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}, \tag{5}$$
$$\boldsymbol{b}_c = \begin{bmatrix} b_{p-1} & \cdots & b_0 \end{bmatrix}, \tag{6}$$
$$\boldsymbol{e} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^\top, \tag{7}$$

with $b_{p-1}, \ldots, b_{q+1} = 0$ and $b_q = 1$ by convention. The roots of $a_c(\cdot)$ and $b_c(\cdot)$ are the process "poles" and "zeros" respectively. A commonly used continuous-time process is the Ornstein-Uhlenbeck (OU) process, which only has a pole at $s = \alpha$ and no zeros. This is stationary for $\alpha < 0$, and is known to be a zero-mean process with weighted-exponential temporal covariance [1]. This can be used as a model for the acceleration of a maneuvering object, giving the Singer model [14].

A discrete-time SSM that has exactly the same properties as a linear SDE can be produced by directly evaluating the discrete state transition matrix and driving covariance, as in [1, page 80]. However, this produces system matrices with complicated structure as each term in the produced matrices is dependent on all model parameters, which makes model parameter inference difficult (see, for example, the Singer model discretisation [14]). Model and process inference becomes much simpler with discrete-time ARMA models due to the simple SSM system matrices. This work shows how these models can be interrelated.

Denote the discrete-time sampling of $x(t)$ by $x[k] \triangleq x(kT_s)$ for $k \in \mathbb{N}_0$, where T_s is the constant sampling period. We say x is an instance of an ARMA(p, q) process if it satisfies:

$$a_d(L)x[k] = b_d(L)\varepsilon[k], \tag{8}$$

where the polynomials a_d and b_d are in terms of a time-lag operator, L, and with ε a white noise process with variance σ_d^2. These can also be written as the output of an SSM in terms of a latent vector process $\boldsymbol{z}_d$ [11]:

$$\boldsymbol{z}_d[t+1] = \boldsymbol{A}_d\boldsymbol{z}_d[t] + \boldsymbol{e}\varepsilon[t], \tag{9}$$
$$x[t] = \boldsymbol{b}\boldsymbol{z}_d[t], \tag{10}$$

with state matrices with the same structure as in (3), but with different coefficients.

III. Process Covariance Matching

It has been shown that all stationary CARMA(p, q) processes can be sampled to produce a valid, stationary ARMA$(p, p-1)$ process with the same temporal covariance [8], [20]. The covariance with time-delay h can be found for both types of processes in terms of their poles and zeros, but it is complicated for general processes. If one assumes there are no repeated poles, then the covariances take the form:

$$\gamma_c(h) = \sigma_c^2 \sum_{i=1}^{p} K_{c,i} \exp(\lambda_i |h|), \tag{11}$$
$$\gamma_d[h] = \sigma_d^2 \sum_{i=1}^{p} K_{d,i} \lambda_i^{|h|+1}, \tag{12}$$

where the $K_{c,i}$ and $K_{d,i}$ can be determined easily. The CARMA process can be sampled at intervals of T_s, providing a discrete covariance function. From this, the power spectral densities of the sampled processes are both rational functions and the poles, zeros and variance of one type of model can be found in terms of another by solving a degree $p-1$ polynomial [9].

These methods can be extended to find corresponding ARMA models for the stationary increments of an integrated CARMA process [12]. However, the resulting transformation is complicated for the simplest case of a once-integrated process. These transformations also do not consider how to transform the latent state between models. In the following sections we provide a numerical algorithm for transforming between two process models by considering the predictive distributions, rather than the process statistics as a whole.

IV. Predictive Distribution Matching

Consider an ARMA process in state-space form as in (5). Let the distribution of the initial states, $\boldsymbol{z}_d[0]$, have mean $\boldsymbol{m}_d$ and covariance $\boldsymbol{P}_d$. The k-step-ahead prediction of the latent state $\boldsymbol{z}$ and the process x are:

$$\boldsymbol{z}_d[k] = \boldsymbol{A}_d^k\boldsymbol{z}_d[0] + \sum_{i=1}^{k} \boldsymbol{A}_d^{i-1}\boldsymbol{e}\varepsilon[k-i], \tag{13}$$
$$x[k] = \boldsymbol{b}_d\boldsymbol{z}_d[k]. \tag{14}$$

The moments of the distribution of $x[k]$ can be determined, with some important terms used in the remaining discussion written as separate functions:

$$\mathbb{E}\{x[k]\} = \boldsymbol{b}_d\boldsymbol{\Theta}_d(k)\boldsymbol{m}_d, \tag{15}$$
$$\text{Cov}\big(x[k]\big) = \boldsymbol{b}_d\Big(\boldsymbol{\Theta}_d(k)\boldsymbol{P}_d\boldsymbol{\Theta}_d(k)^\top + \boldsymbol{K}_d(k)\Big)\boldsymbol{b}_d^\top, \tag{16}$$
$$\boldsymbol{\Theta}_d(k) \triangleq \boldsymbol{A}_d^k, \tag{17}$$
$$\boldsymbol{K}_d(k) \triangleq \sigma_d^2 \sum_{i=1}^{k} \boldsymbol{A}_d^{i-1}\boldsymbol{e}\boldsymbol{e}^\top(\boldsymbol{A}_d^\top)^{i-1}. \tag{18}$$

The same approach is applied to the CARMA process, using known formulas for the predictive distributions of a linear SDE from [1, page 80]. Once again writing some terms separately, we get the predictive mean and covariance of a CARMA SSM as in (3) as:

$$\mathbb{E}\big\{x(kT_s)\big\} = \boldsymbol{b}_c\boldsymbol{\Theta}_c(k)\boldsymbol{m}_c, \tag{19}$$

$$\mathrm{Cov}\big(x(kT_s)\big) = \boldsymbol{b}_c\Big(\boldsymbol{\Theta}_c(k)\boldsymbol{P}_c\boldsymbol{\Theta}_c(k)^\top + \boldsymbol{K}_c(k)\Big)\boldsymbol{b}_c^\top, \tag{20}$$

$$\boldsymbol{\Theta}_c(k) \triangleq \exp(\boldsymbol{A}_cT_sk), \tag{21}$$

$$\boldsymbol{K}_c(k) \triangleq \sigma_c^2\int_0^{kT_s}\exp(\boldsymbol{A}_c(kT_s-\tau))\boldsymbol{e} \cdot \boldsymbol{e}^\top\exp(\boldsymbol{A}_c(kT_s-\tau))^\top d\tau. \tag{22}$$

The algorithm developed below takes a known model, and finds parameters for a “conjugate” model consistent for (15) and (19), as well as (16) and (20), equal for some values of k. It is not always possible to make them identically equal for all k, so here the error is minimised over $k \in \{1, 2, \dots, p\}$.

A. Transforming the model

The predictive statistics of ARMA and CARMA processes have expressions that take almost identical forms, so the following discussion for switching between ARMA and CARMA processes applies just as well to the converse. As a result, the discussion to follow will use un-subscripted symbols to denote the model for which all the parameters are known, and a starred subscript will represent the corresponding quantity in the conjugate model. For example: if the ARMA system is known, then the conjugate is CARMA, with $\boldsymbol{b} \triangleq \boldsymbol{b}_d$ and $\boldsymbol{b}_* \triangleq \boldsymbol{b}_c$.

From (11) and (12), we see that the covariances of two processes can only be equal for all h if the bases of the powers of h are equal. $\boldsymbol{A}_*$ can be found easily by exchanging poles, λ, through:

$$\lambda_{d,i} = \exp(\lambda_{c,i}T_s). \tag{23}$$

This mapping is not unique, so we use the convention $|\mathrm{Im}(\lambda_{c,i})| < \pi$.

We now find the MA terms of the conjugate model. Consider the case of having no uncertainty around the initial states for either model: $\boldsymbol{P} = \boldsymbol{0} = \boldsymbol{P}_*$. Equating the covariance equations, (16) and (20), thus reduces to:

$$\boldsymbol{b}\boldsymbol{K}(k)\boldsymbol{b}^\top = \boldsymbol{b}_*\boldsymbol{K}_*(k)\boldsymbol{b}_*^\top \tag{24}$$

for $k \in \{1, \dots, p\}$. This can be written as a system of quadratic vector equations which will be re-used in a following discussion:

$$\boldsymbol{v}^\top\boldsymbol{M}_i\boldsymbol{v} = r_i \tag{25}$$

for $i \in \{1, \dots, p\}$. A least-squares solution to this system uses loss, gradient, and Hessian:

$$E(\boldsymbol{v}) \triangleq \sum_{i=1}^{p}(\boldsymbol{v}^\top\boldsymbol{M}_i\boldsymbol{v} - r_i)^2, \tag{26}$$

$$\boldsymbol{G}(\boldsymbol{v}) = 4\sum_{i=1}^{p}(\boldsymbol{v}^\top\boldsymbol{M}_i\boldsymbol{v} - r_i)\boldsymbol{M}_i\boldsymbol{v}, \tag{27}$$

$$\boldsymbol{H}(\boldsymbol{v}) = 4\sum_{i=1}^{p}\left((\boldsymbol{v}^\top\boldsymbol{M}_i\boldsymbol{v} - r_i)\boldsymbol{M}_i + 2\boldsymbol{M}_i\boldsymbol{v}\boldsymbol{v}^\top\boldsymbol{M}_i^\top\right). \tag{28}$$

Algorithm 1 Procedure for constructing the moving average component, $\boldsymbol{b}_*$, and the driving variance, σ_*^2, of the conjugate model.

```
Require: $\boldsymbol{A}$, $\boldsymbol{b}$, $\sigma^2$, $T_s$
Construct $\boldsymbol{A}_*$ from (5) and (23)
Evaluate $\boldsymbol{M}_i \triangleq \boldsymbol{K}_*(i)$ for $i \in \{1, \dots, p\}$
Evaluate $r_i \triangleq \boldsymbol{b}\boldsymbol{M}(i)\boldsymbol{b}^\top$ for $i \in \{1, \dots, p\}$
Define $E(\boldsymbol{v})$, $\boldsymbol{G}(\boldsymbol{v})$, and $\boldsymbol{H}(\boldsymbol{v})$ as in (26), (27), (28)
Initialise $\boldsymbol{v} \leftarrow \boldsymbol{v}^{(0)}$
while $E(\boldsymbol{v}) > E_{\text{min}}$ do
    $\boldsymbol{v} \leftarrow \boldsymbol{v} - \boldsymbol{H}^{-1}\boldsymbol{G}$
end while
$s \leftarrow \mathrm{FirstNonZero}(\boldsymbol{v})$
$\boldsymbol{b}_* \leftarrow \frac{\boldsymbol{v}}{s}$
$\sigma_*^2 \leftarrow s^2\sigma^2$
return $\boldsymbol{b}_*$, $\sigma_*^2$
```

The Newton-Raphson [21] algorithm provides iterative estimates of $\boldsymbol{v}$ which may be repeated until the loss is sufficiently small (here 1×10^{-12}):

$$\boldsymbol{v}^{(i+1)} = \boldsymbol{v}^{(i)} - \boldsymbol{H}\big(\boldsymbol{v}^{(i)}\big)^{-1}\boldsymbol{G}(\boldsymbol{v}^{(i)}). \tag{29}$$

The MA coefficients of the conjugate model are obtained by setting $\boldsymbol{M}_i = \boldsymbol{K}_*(i)$ and $r_i = \boldsymbol{b}\boldsymbol{K}(i)\boldsymbol{b}^\top$ so the iteration converges to $\boldsymbol{v} = \boldsymbol{b}_*$. Importantly, (25) is symmetric up to the sign of $\boldsymbol{v}$. We can divide through by the first non-zero term, and scale the original variance by this leading term squared to provide the conjugate variance and ensure our $\boldsymbol{b}_*$ has unit leading coefficient. The gradient in (27) vanishes at $\boldsymbol{v} = \boldsymbol{0}$, so special care should be taken to initialise the iteration away from this point. In practice, the Hessian was found to be non-singular although no theoretical guarantees are available in this work. The AR and MA parameters, as well as the variance of the conjugate model can now be determined; this is summarised in Algorithm 1.

B. Transforming the initial state

The conjugate latent state mean and covariance can be determined once the conjugate model parameters are known. In order to find the mean, equate (15) and (19):

$$\boldsymbol{b}\boldsymbol{\Theta}(k)\boldsymbol{m} = \boldsymbol{b}_*\boldsymbol{\Theta}_*(k)\boldsymbol{m}_* \tag{30}$$

for $k \in \{1, \dots, p\}$. The latent mean follows as:

$$\boldsymbol{m}_* = \begin{bmatrix}\boldsymbol{b}_*\boldsymbol{\Theta}_*(1)\\ \vdots \\ \boldsymbol{b}_*\boldsymbol{\Theta}_*(p)\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{b}\boldsymbol{\Theta}(1)\boldsymbol{m}\\ \vdots \\ \boldsymbol{b}\boldsymbol{\Theta}(p)\boldsymbol{m}\end{bmatrix}. \tag{31}$$

The inverse here always exists as the matrix has rank p when the polynomials $a_*(\cdot)$ and $b_*(\cdot)$ do not share roots, due to Hautus’ lemma [22, Theorem 15.9].

The covariance can be determined by equating (16) and (20). The terms involving $\boldsymbol{K}(k)$ are equal since that is exactly the condition that $\boldsymbol{b}_*$ satisfies. We thus get:

$$\boldsymbol{b}_*\Theta(k)_*\boldsymbol{P}_*\Theta(k)_*^\top\boldsymbol{b}_*^\top = \boldsymbol{b}\Theta(k)\boldsymbol{P}\Theta(k)^\top\boldsymbol{b}^\top. \tag{32}$$

Algorithm 2 Procedure for transforming the initial latent state distribution between known conjugate models.

```
Require: $\boldsymbol{A}$, $\boldsymbol{b}$, $\sigma^2$, $\boldsymbol{A}_*$, $\boldsymbol{b}_*$, $\sigma_*^2$, $T_s$, $\boldsymbol{m}$, $\boldsymbol{P}$
Evaluate $\Theta(i)$ and $\Theta_*(i)$ for $i \in \{1, \dots, p\}$
Evaluate $\boldsymbol{m}_*$ from (31)
Evaluate $\boldsymbol{v}_i^\top \triangleq \boldsymbol{b}_*\Theta_*(i)$ for $i \in \{1, \dots, p\}$
Evaluate $r_i \triangleq \boldsymbol{b}\Theta(i)\boldsymbol{P}\Theta(i)^\top \boldsymbol{b}^\top$ for $i \in \{1, \dots, p\}$
Define $E(\boldsymbol{M})$, $\boldsymbol{G}(\boldsymbol{M})$, and $\boldsymbol{H}(\boldsymbol{M})$ as in (34), (35), (36)
Initialise $\boldsymbol{M} \leftarrow \boldsymbol{0}$
while $E(\boldsymbol{M}) > E_{\text{min}}$ do
    $\boldsymbol{M} \leftarrow \boldsymbol{M} - \boldsymbol{H}^{-1}\boldsymbol{G}$
end while
$\boldsymbol{P}_* \leftarrow \frac{1}{2}(\boldsymbol{M} + \boldsymbol{M}^\top)$
return $\boldsymbol{m}_*$, $\boldsymbol{P}_*$
```

This is solved through another least squared-error estimate of a system in the form of (25), however with the variable of interest being a constant $\boldsymbol{M}$:

$$\boldsymbol{v}_i^\top \boldsymbol{M} \boldsymbol{v}_i = r_i. \tag{33}$$

Defining the loss similarly to before:

$$E(\boldsymbol{M}) \triangleq \sum_{i=1}^{p} (\boldsymbol{v}_i^\top \boldsymbol{M} \boldsymbol{v}_i - r_i)^2, \tag{34}$$

$$G(\boldsymbol{M}) = 2\sum_{i=1}^{p} (\boldsymbol{v}_i^\top \boldsymbol{M} \boldsymbol{v}_i - r_i)\boldsymbol{v}_i \boldsymbol{v}_i^\top, \tag{35}$$

$$H(\boldsymbol{M}) = 2\sum_{i=1}^{p} (\boldsymbol{v}_i \boldsymbol{v}_i^\top)(\boldsymbol{v}_i \boldsymbol{v}_i^\top)^\top. \tag{36}$$

The Hessian in (36) is a positive semi-definite constant if some $\boldsymbol{v}_i \neq \boldsymbol{0}$, so the inverse always exists. The conjugate covariance can be found by letting $\boldsymbol{v}_i^\top = \boldsymbol{b}_*\Theta(k)_*$ and taking r_i as the right side of (32). This method does not guarantee that the solution produced, $\boldsymbol{M}$, is a valid covariance matrix. It can be made symmetric by noting that if $\boldsymbol{M}$ is a solution, then so are both $\boldsymbol{M}^\top$ and $\frac{1}{2}(\boldsymbol{M} + \boldsymbol{M}^\top)$, the latter being symmetric. The gradient is non-zero for $\boldsymbol{M} = \boldsymbol{0}$, so the iteration can be initialised at this point. This approach is summarised in Algorithm 2.

The update steps in both algorithms will be $\mathcal{O}(p^4)$ as the inverse Hessian is evaluated at p points. This is not unreasonable, as p is usually not large, and further hardware parallelisation will produce better performance in practice. Further study may provide guarantees for number of iterations to perform, but empirically it was found to be low.

V. Equivalence Examples

We demonstrate the algorithms developed here by first comparing them to the results produced by [9] on strictly stationary processes, as well as conjugate state predictions for a stationary process. Thereafter, we show the non-trivial conjugate model parameters experimentally for some common classes of non-stationary models.

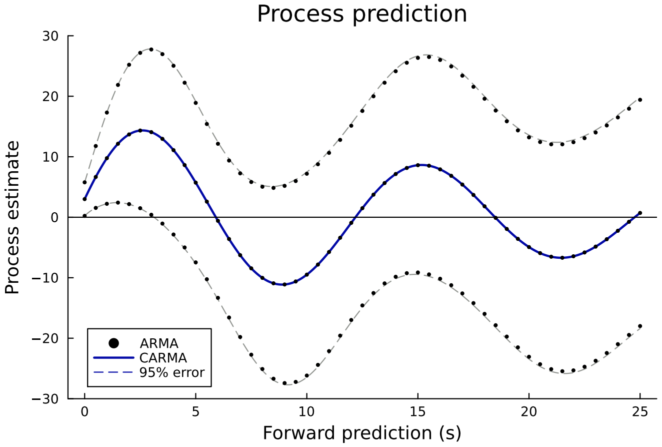

Fig. 1 Forward-predictions of the distribution for x using both the original ARMA model with initial latent distribution and the conjugate CARMA model and its calculated initial latent distribution..

A. Stationary processes

Consider a stationary CARMA process given by:

$$a_c(s) = s^2 + 1.5s + 0.5, \tag{37}$$

$$b_c(s) = s + 0.25, \quad \sigma_c^2 = 1. \tag{38}$$

With $T_s = 1\,$s, our algorithm provides the conjugate model:

$$a_d(z) = z^2 - 1.856z + 0.861, \tag{39}$$

$$b_d(z) = z - 0.975, \quad \sigma_d^2 = 0.0884, \tag{40}$$

which matches that found in [9, Example 3.11]. We also successfully reconstruct the original CARMA model when finding the conjugate of the derived model. The state transformation algorithm is demonstrated for an AR process with the following parameters and latent state distribution:

$$a_d(z) = z^2 - 1.899z + 0.960, \tag{41}$$

$$b_d(z) = z^2, \quad \sigma_d^2 = 0.5, \tag{42}$$

$$\boldsymbol{m} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}, \quad \boldsymbol{P} = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 1 \end{bmatrix}. \tag{43}$$

With $T_s = 0.5\,$s, the conjugate system produced is:

$$a_c(s) = s^2 + 0.081s + 0.251, \tag{44}$$

$$b_c(s) = s + 4.863, \quad \sigma_c^2 = 0.210, \tag{45}$$

$$\boldsymbol{m}_* = \begin{bmatrix} 1.740 \\ 0.276 \end{bmatrix}, \quad \boldsymbol{P}_* = \begin{bmatrix} 0.378 & 0.060 \\ 0.060 & 0.052 \end{bmatrix}. \tag{46}$$

Using the initial state as above, the predictive state distribution can be evaluated. Figure 1 shows that the ARMA and CARMA predictive distributions are very close for the 25 s shown, so the conjugate CARMA system is a good approximation.

B. Non-stationary processes

In tracking literature, it is common to integrate the OU process some number of times to produce maneuvering object dynamics. These are examples of CAR processes, so they can

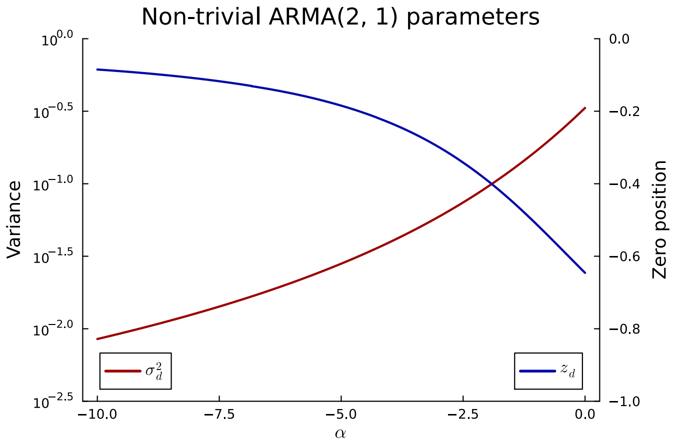

Fig. 2 ARMA zero and variance for a once-integrated OU process in terms of parameter α with $\sigma_c^2 = 1$ and $T_s = 1\,\mathrm{s}$.

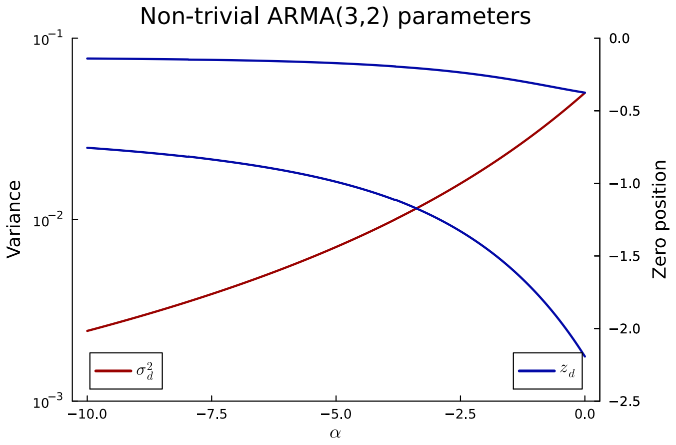

Fig. 3 ARMA zeros and variance for the Singer process in terms of parameter α with $\sigma_c^2 = 1$ and $T_s = 1\,\mathrm{s}$.

be approximated by discrete time ARMA processes. It is well-known that the OU process is equivalent to an AR(1) process; so a naive approach represents the n-times integrated process by an AR$(1+n)$ process with n poles at $z = 1$. However, Figures 2 and 3 show that the zeros and variance of an ARMA representation are dependent on the OU parameter, α, for a once-integrated, and twice-integrated OU process respectively.

VI. Conclusion

This work has shown numerical algorithms for finding equivalent ARMA and CARMA models and latent state distributions in terms of short-term predictive distributions. These correspond with existing literature on stationary processes, and can be extended to non-stationary processes without further modifications. This can be applied to object tracking algorithms using sensors with different sampling rates. Future work will focus on the accuracy of the produced non-stationary models and provide performance guarantees for the proposed algorithms. Extensions to non-linear or heavy-tailed systems can also be studied.

Appendix A
Code repository

An implementation of the proposed algorithm, and the experiments performed here can be found in the following repository:

github.com/RalphMcDougall/EUSIPCO-2025-CARMA

References

[1] S. Sarkka, *Applied stochastic differential equations*, ser. Institute of Mathematical Statistics Textbooks, A. Solin, Ed. Cambridge University Press, 2019.

[2] A. R. Bergstrom, *Continuous time econometric modelling*. Oxford University Press, 1990.

[3] M. C. Mariani and O. K. Tweneboah, "Stochastic differential equations applied to the study of geophysical and financial time series," *Physica A: Statistical Mechanics and its Applications*, vol. 443, pp. 170–178, Feb. 2016.

[4] B. K. Oksendal, *Stochastic differential equations : an introduction with applications*, 6th ed. Berlin: Berlin, 2005.

[5] A. Harvey and J. H. Stock, "Continuous time autoregressive models with common stochastic trends," *Journal of Economic Dynamics and Control*, vol. 12, no. 2, pp. 365–384, Jun. 1988.

[6] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, 1st ed. USA: John Wiley & Sons, Inc., 1996.

[7] A. C. Harvey, *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1990.

[8] A. W. Phillips, "The Estimation of Parameters in Systems of Stochastic Differential Equations," *Biometrika*, vol. 46, no. 1/2, pp. 67–76, 1959.

[9] P. J. Brockwell and A. Lindner, "Sampling, Embedding and Inference for CARMA Processes," *Journal of Time Series Analysis*, vol. 40, no. 2, pp. 163–181, Mar. 2019, publisher: John Wiley & Sons, Ltd.

[10] A. E. Brockwell and P. J. Brockwell, "A Class of Non-Embeddable ARMA Processes," *Journal of Time Series Analysis*, vol. 20, no. 5, pp. 483–486, Sep. 1999, publisher: John Wiley & Sons, Ltd.

[11] P. J. Brockwell, *Time series : theory and methods*, 2nd ed., ser. Springer Series in Statistics, R. Davis, Ed. New York: Springer Science & Business Media, 1996.

[12] P. Brockwell and A. Lindner, "Integration of CARMA processes and spot volatility modelling," *Journal of Time Series Analysis*, vol. 34, no. 2, pp. 156–167, Mar. 2013, publisher: John Wiley & Sons, Ltd.

[13] F. Lydeard, B. I. Ahmad, and S. Godsill, "Integrated Gaussian Processes for Tracking," *IEEE Open Journal of Signal Processing*, vol. 6, pp. 99–107, 2025.

[14] R. A. Singer, "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-6, no. 4, pp. 473–483, Jul. 1970.

[15] R. J. McDougall and S. J. Godsill, "Target Tracking Using a Time-Varying Autoregressive Dynamic Model," *IEEE Open Journal of Signal Processing*, vol. 6, pp. 147–155, 2025.

[16] S. J. Godsill, "Bayesian Enhancement of Speech and Audio Signals which can be Modelled as ARMA Processes," *International Statistical Review*, vol. 65, no. 1, pp. 1–21, Apr. 1997, publisher: John Wiley & Sons, Ltd.

[17] M. Arnold, X. Milner, H. Witte, R. Bauer, and C. Braun, "Adaptive AR modeling of nonstationary time series by means of Kalman filtering," *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 5, pp. 553–562, May 1998.

[18] G. Barnett, G. Barnett, R. Kohn, and S. Sheather, "Bayesian estimation of an autoregressive model using Markov chain Monte Carlo," *Journal of Econometrics.*, vol. 74, no. 2, pp. 237–254, 1996, number: 2 Place: Amsterdam : Publisher: North-Holland.

[19] R. Cont, *Financial modelling with jump processes*, P. Tankov, Ed. Boca Raton, Fla.: Boca Raton, Fla., 2004.

[20] P. Brockwell, "Continuous-time ARMA processes," in *Handbook of Statistics*. Elsevier, Jan. 2001, vol. 19, pp. 249–276.

[21] C. M. Bishop, *Pattern recognition and machine learning*, ser. Information science and statistics. Springer Science + Business Media, 2006.

[22] J. P. Hespanha, *Linear Systems Theory*, 2nd ed. Princeton University Press, 2018.