

## EJERCICIOS TEÓRICOS

**Ejercicio 1.** Kalman Gain

Sabemos que el estado predicho y la observación se modelan mediante las distribuciones normales  $\mathcal{N}(x; \bar{\mu}, \bar{\sigma}^2)$  y  $\mathcal{N}(z; x, \sigma_{obs}^2)$ , respectivamente. Sean

$$f(x) = \frac{1}{\bar{\sigma}\sqrt{2\pi}} e^{-\frac{(x-\bar{\mu})^2}{2\bar{\sigma}^2}}$$

y

$$g(z) = \frac{1}{\sigma_{obs}\sqrt{2\pi}} e^{-\frac{(z-x)^2}{2\sigma_{obs}^2}}$$

sus correspondientes funciones de densidad de probabilidad. Luego, su multiplicación resulta

$$(1) \quad f(x)g(z) = \frac{1}{\bar{\sigma}\sigma_{obs}2\pi} e^{-\frac{(x-\bar{\mu})^2\sigma_{obs}^2 + (z-x)^2\bar{\sigma}^2}{2\bar{\sigma}^2\sigma_{obs}^2}}$$

Concentrémonos en el numerador del exponente. Vemos que

$$(2) \quad \begin{aligned} (x - \bar{\mu})^2\sigma_{obs}^2 + (z - x)^2\bar{\sigma}^2 &= x^2(\sigma_{obs}^2 + \bar{\sigma}^2) - 2x(\bar{\mu}\sigma_{obs}^2 + z\bar{\sigma}^2) + \bar{\mu}^2\sigma_{obs}^2 + z^2\bar{\sigma}^2 \\ &= (\sigma_{obs}^2 + \bar{\sigma}^2) \left( x - \frac{\bar{\mu}\sigma_{obs}^2 + z\bar{\sigma}^2}{\sigma_{obs}^2 + \bar{\sigma}^2} \right)^2 - \frac{(\bar{\mu}\sigma_{obs}^2 + z\bar{\sigma}^2)^2}{\sigma_{obs}^2 + \bar{\sigma}^2} + \bar{\mu}^2\sigma_{obs}^2 + z^2\bar{\sigma}^2 \\ &= (\sigma_{obs}^2 + \bar{\sigma}^2) (x - \mu)^2 + \sigma^2 (z - \bar{\mu})^2 \end{aligned}$$

donde en el primer paso se expandieron los binomios al cuadrado, en el segundo paso se completó cuadrados para la variable  $x$  y en el último paso se emplearon las definiciones de  $\mu := \bar{\mu} + K(z - \bar{\mu})$  y  $\sigma^2 := (1 - K)\bar{\sigma}^2$ . Reemplazando en (1), tenemos que

$$(3) \quad \begin{aligned} f(x)g(z) &= \frac{1}{\bar{\sigma}\sigma_{obs}2\pi} e^{-\frac{(\sigma_{obs}^2 + \bar{\sigma}^2)(x-\mu)^2 + \sigma^2(z-\bar{\mu})^2}{2\bar{\sigma}^2\sigma_{obs}^2}} \\ &= \frac{1}{\sqrt{\sigma_{obs}^2 + \bar{\sigma}^2}\sqrt{2\pi}} e^{-\frac{(z-\bar{\mu})^2}{2(\sigma_{obs}^2 + \bar{\sigma}^2)}} \cdot \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \end{aligned}$$

que es proporcional a la función de densidad de probabilidad de una distribución normal  $\mathcal{N}(x; \mu, \sigma^2)$ .

**Ejercicio 2.** Jacobiano del Modelo de Movimiento

Comenzamos calculando el Jacobiano respecto del estado  $G$ . Tenemos las siguientes derivadas parciales:

$$(4) \quad \begin{aligned} \frac{\partial x'}{\partial x} &= 1, & \frac{\partial x'}{\partial y} &= 0, & \frac{\partial y'}{\partial x} &= 0, & \frac{\partial y'}{\partial y} &= 1, & \frac{\partial \theta'}{\partial x} &= 0, & \frac{\partial \theta'}{\partial y} &= 0, & \frac{\partial \theta'}{\partial \theta} &= 1, \\ \frac{\partial x'}{\partial \theta} &= \frac{\partial}{\partial \theta} \delta_{trans} \cos(\theta_t + \delta_{rot1}) = -\delta_{trans} \sin(\theta_t + \delta_{rot1}), \\ \frac{\partial y'}{\partial \theta} &= \frac{\partial}{\partial \theta} \delta_{trans} \sin(\theta_t + \delta_{rot1}) = \delta_{trans} \cos(\theta_t + \delta_{rot1}). \end{aligned}$$

Por lo tanto, nos queda

$$(5) \quad G = \begin{bmatrix} 1 & 0 & -\delta_{trans} \sin(\theta_t + \delta_{rot1}) \\ 0 & 1 & \delta_{trans} \cos(\theta_t + \delta_{rot1}) \\ 0 & 0 & 1 \end{bmatrix}.$$

Luego, calculamos el Jacobiano con respecto al control  $V$ . Las derivadas parciales resultan:

$$(6) \quad \begin{aligned} \frac{\partial x'}{\partial \delta_{rot2}} &= 0, & \frac{\partial y'}{\partial \delta_{rot2}} &= 0, & \frac{\partial \theta'}{\partial \delta_{rot1}} &= 1, & \frac{\partial \theta'}{\partial \delta_{trans}} &= 0, & \frac{\partial \theta'}{\partial \delta_{rot2}} &= 1, \\ \frac{\partial x'}{\partial \delta_{rot1}} &= \frac{\partial}{\partial \delta_{rot1}} \delta_{trans} \cos(\theta_t + \delta_{rot1}) = -\delta_{trans} \sin(\theta_t + \delta_{rot1}) \\ \frac{\partial x'}{\partial \delta_{trans}} &= \frac{\partial}{\partial \delta_{trans}} \delta_{trans} \cos(\theta_t + \delta_{rot1}) = \cos(\theta_t + \delta_{rot1}) \\ \frac{\partial y'}{\partial \delta_{rot1}} &= \frac{\partial}{\partial \delta_{rot1}} \delta_{trans} \sin(\theta_t + \delta_{rot1}) = \delta_{trans} \cos(\theta_t + \delta_{rot1}) \\ \frac{\partial y'}{\partial \delta_{trans}} &= \frac{\partial}{\partial \delta_{trans}} \delta_{trans} \sin(\theta_t + \delta_{rot1}) = \sin(\theta_t + \delta_{rot1}). \end{aligned}$$

Entonces, nos queda el Jacobiano

$$(7) \quad V = \begin{bmatrix} -\delta_{trans} \sin(\theta_t + \delta_{rot1}) & \cos(\theta_t + \delta_{rot1}) & 0 \\ \delta_{trans} \cos(\theta_t + \delta_{rot1}) & \sin(\theta_t + \delta_{rot1}) & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

## EJERCICIOS PRÁCTICOS

**NOTA:** El código al que refiere esta sección se encuentra en <https://github.com/doraemon96/RoboticaMovil2023/tree/master/TP4>

### Ejercicio 3. Implementar EKF

Para la implementación del EKF se completaron los métodos `Field.G`, `Field.V` y `Field.H`, definidos en `soccer_field.py`. Los dos primeros computan los Jacobianos de la dinámica con respecto al estado y al control, respectivamente, mientras que el último calcula el Jacobiano de la observación con respecto al estado. Además, se implementó el algoritmo del EKF en el método `ExtendedKalmanFilter.update`, definido en `ekf.py`.

a) Ejecutando el EKF con los parámetros predeterminados y semilla igual a cero obtenemos el camino del robot real y el camino estimado por el filtro graficados en la Fig. 1. Se obtuvo un error medio de posición (MPE) de aproximadamente 8,99 y un ANEES de aproximadamente 1,47.

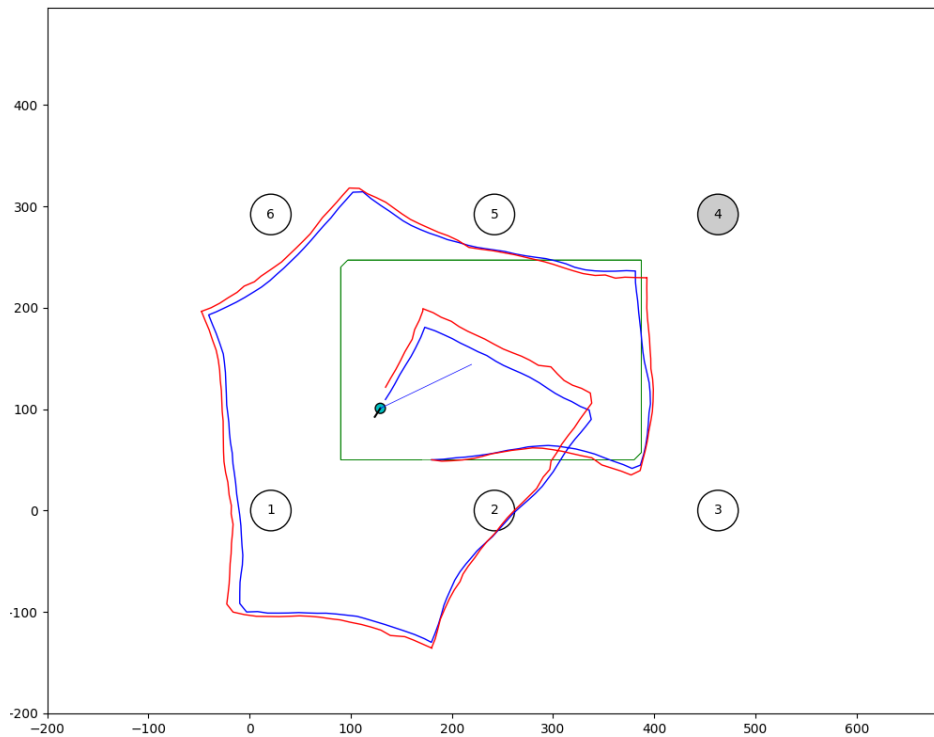


FIGURA 1. Camino del robot real (azul) y camino estimado por el EKF (rojo) con parámetros predeterminados (y  $\text{seed} = 0$ ).

b) Se simuló el sistema multiplicando los factores  $\alpha$  y  $\beta$  por los valores  $r := [1/64, 1/16, 1/4, 4, 16, 64]$ , realizando 10 ejecuciones para cada valor de  $r$ . Para ello, se modificó el archivo `localization.py` de manera de permitir la ejecución automática de todas las corridas. En la Fig. 2 se muestra la media del MPE obtenido en las 10 ejecuciones para cada valor de  $r$ . Podemos apreciar que el MPE crece a medida que las varianzas de los ruidos de los modelos de odometría y observación aumentan (equivalente a aumentar  $\alpha$  y  $\beta$ ).

Es claro que al sistema no le es posible compensar un error con una varianza muy grande por más que el modelo también lo contemple, lo cual puede deberse a que el sensor exteroceptivo (en este caso el sensor de sólo ángulo) no provee suficiente información al sistema. Una posible suposición es que al proveer el ángulo hacia un único landmark en cada instante no está pudiendo hacer converger al sistema ante errores de gran magnitud.

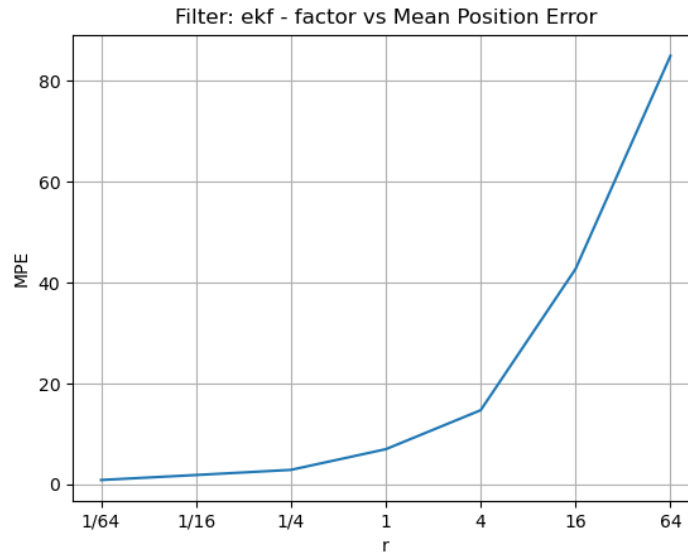


FIGURA 2. Valor medio del MPE en 10 corridas para cada valor de  $r$ .

c) En este ejercicio, se multiplicaron los factores  $\alpha$  y  $\beta$  que emplea el EKF por los valores de  $r$ , pero a diferencia del ejercicio anterior, se mantuvieron los valores por defecto de los factores que generan los datos. Nuevamente se realizaron 10 ejecuciones para cada valor de  $r$ . En la Fig. 3 se muestra la media del MPE obtenido en las 10 ejecuciones para cada valor de  $r$ , mientras que en la Fig. 4 se muestra la media del ANEES obtenido en las 10 ejecuciones para cada valor de  $r$ .

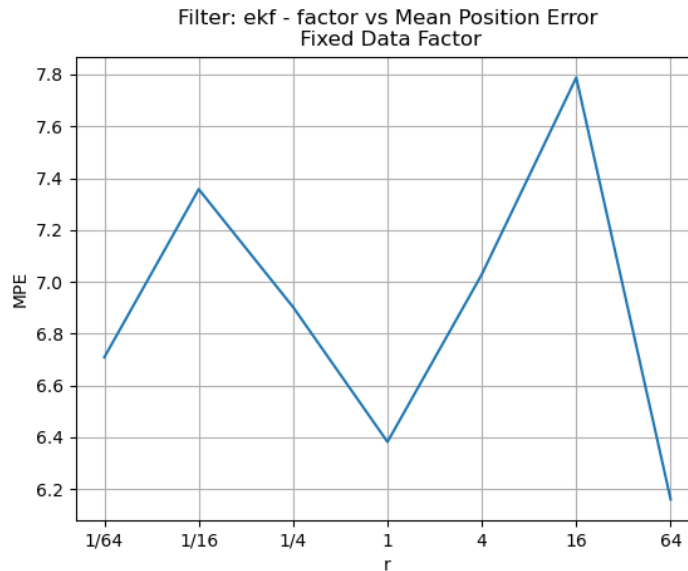


FIGURA 3. Valor medio del MPE en 10 corridas para cada valor de  $r$  con factores de generación de datos fijos.

Podemos notar que el MPE se redujo con respecto a lo visto en el ejercicio anterior cuando las varianzas empleadas en el filtro fueron mayores a las varianzas de la generación de datos (casos con  $r > 1$ ). De igual manera, se observa que el MPE incrementó cuando las varianzas empleadas en el filtro fueron menores a las varianzas de la generación de datos (casos con  $r < 1$ ). En consecuencia, podríamos decir que sobrestimar las varianzas de los errores en el filtro conduce a obtener un menor MPE.

En cuanto al ANEES, podemos observar un comportamiento similar: cuando las varianzas empleadas en el filtro son mayores a las varianzas de la generación de datos (casos con  $r > 1$ ) el ANEES es pequeño, mientras que al emplear varianzas en el filtro menores a las varianzas de la generación de datos (casos con  $r < 1$ ) el ANEES es mayor. Un valor alto de ANEES indica que la estima del error es menor al

que realmente existe (i.e., se está subestimando el error). Por otro lado, un valor bajo indica que se está sobreestimando el error.

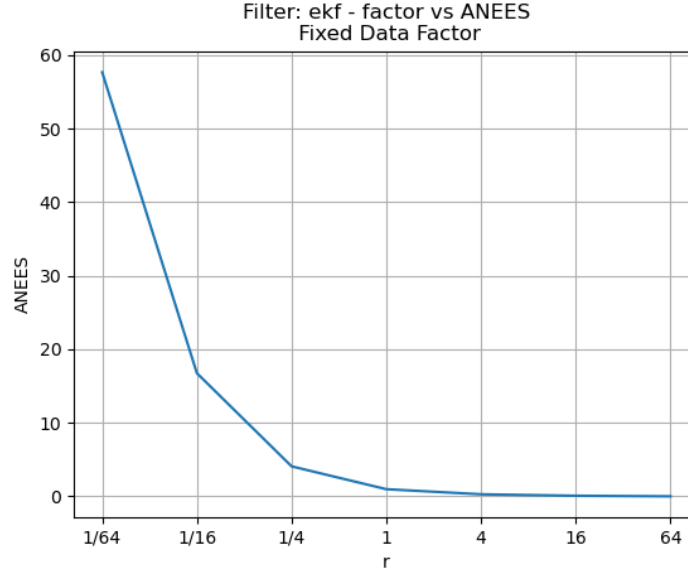


FIGURA 4. Valor medio del ANEES en 10 corridas para cada valor de  $r$  con factores de generación de datos fijos.

De ambos resultados podemos concluir que sobrestimar las varianzas disminuye o mantiene aproximadamente similar la estima del estado del sistema a la que se obtiene con el sistema correctamente modelado, sin embargo aumenta las varianzas de la estima, con lo cual se pierde confianza en la estima del sistema dada por el método.

#### Ejercicio 4. Implementar PF

Se implementó el algoritmo del filtro de partículas modificando los métodos `ParticleFilter.update` y `ParticleFilter.resample` ubicados en `pf.py`.

a) Ejecutando el PF con los parámetros predeterminados y semilla igual a cero obtenemos el camino del robot real y el camino estimado por el filtro graficados en la Fig. 5. Se obtuvo un error medio de posición de aproximadamente 8,57 y un ANEES de 4,91.

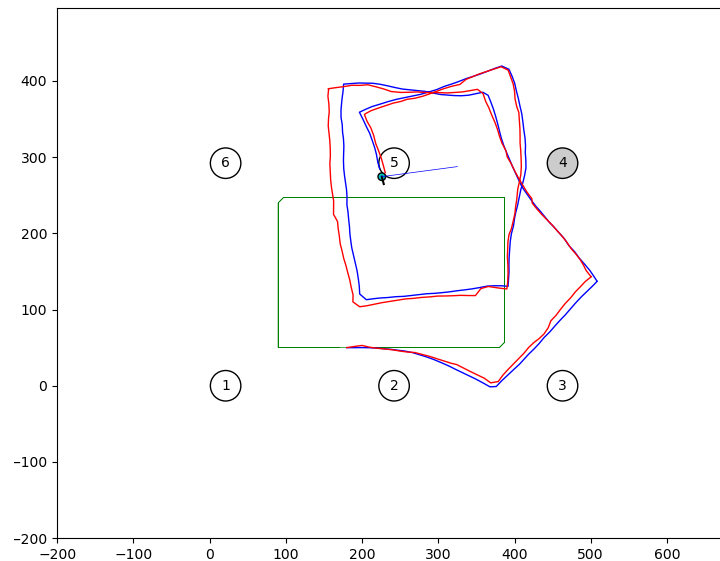


FIGURA 5. Camino del robot real (azul) y camino estimado por el PF (rojo) con parámetros predeterminados (y  $\text{seed} = 0$ ).

b) Se simuló el sistema multiplicando los factores  $\alpha$  y  $\beta$  por los valores  $r := [1/64, 1/16, 1/4, 1, 4, 16, 64]$ , realizando 10 ejecuciones para cada valor de  $r$ . En la Fig. 6 se muestra la media del MPE obtenido en las 10 ejecuciones para cada valor de  $r$ . Podemos notar que, al igual que en el EKF, el MPE crece a medida que las varianzas de los ruidos de los modelos de odometría y observación aumentan.

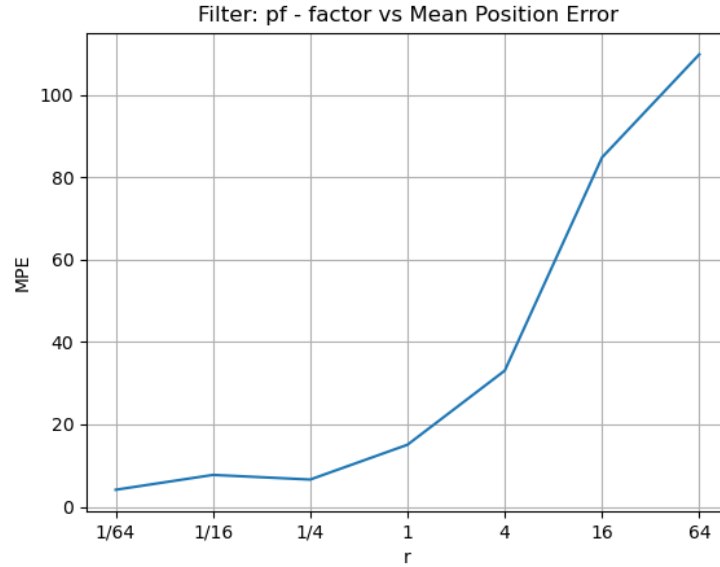


FIGURA 6. Valor medio del MPE en 10 corridas para cada valor de  $r$ .

c) Al igual que lo realizado con el EKF, se multiplicaron los factores  $\alpha$  y  $\beta$  que emplea el PF por los valores de  $r$  mientras que se mantuvieron los valores por defecto de los factores que generan los datos. En la Fig. 7 se muestra la media del MPE obtenido en las 10 ejecuciones para cada valor de  $r$ , mientras que en la Fig. 8 se muestra la media del ANEES obtenido en las 10 ejecuciones para cada valor de  $r$  y en la Fig 9 se muestra un acercamiento a los resultados del ANEES para  $r > 1/4$ .

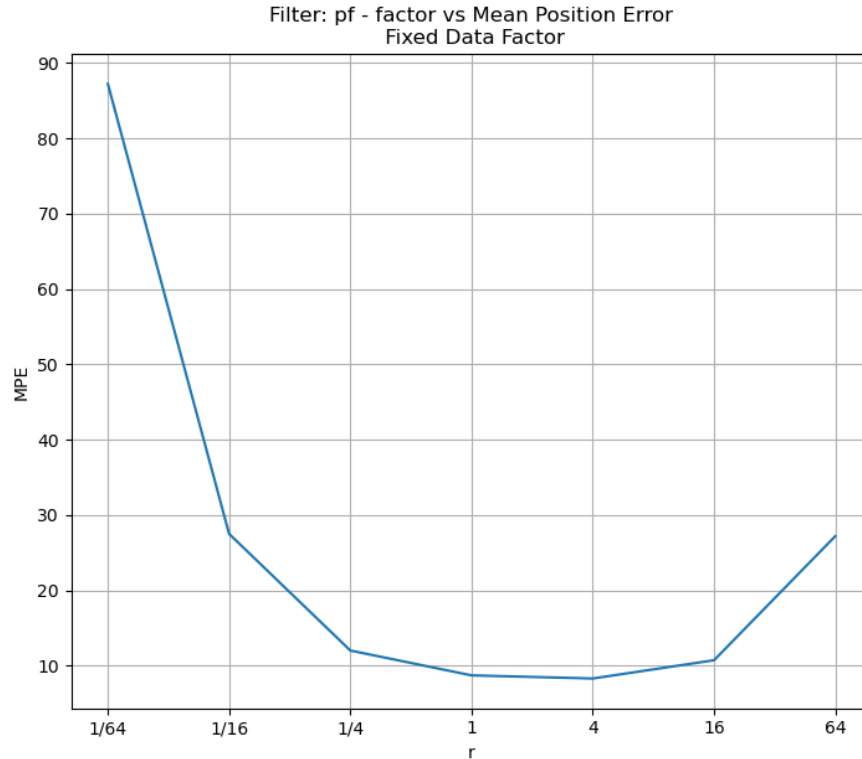


FIGURA 7. Valor medio del MPE en 10 corridas para cada valor de  $r$  con factores de generación de datos fijos.

Al igual que ocurría en EKF, el MPE se reduce con respecto a lo visto en el ejercicio anterior cuando las varianzas empleadas en el PF son mayores a las varianzas de la generación de datos (casos con  $r > 1$ ), y aumenta cuando las varianzas empleadas en el PF son menores a las varianzas de la generación de datos (casos con  $r < 1$ ).

En cuanto al ANEES, podemos observar el mismo comportamiento que mostraba EKF: cuando las varianzas empleadas en el filtro son mayores a las varianzas de la generación de datos (casos con  $r > 1$ ) el ANEES es pequeño, mientras que al emplear varianzas en el filtro menores a las varianzas de la generación de datos (casos con  $r < 1$ ) el ANEES es mayor. Además, para valores pequeños de  $r$  la matriz de covarianza resultó frecuentemente no invertible, por lo que los resultados obtenidos para el ANEES para dichos valores de  $r$  no son representativos del error real del método.

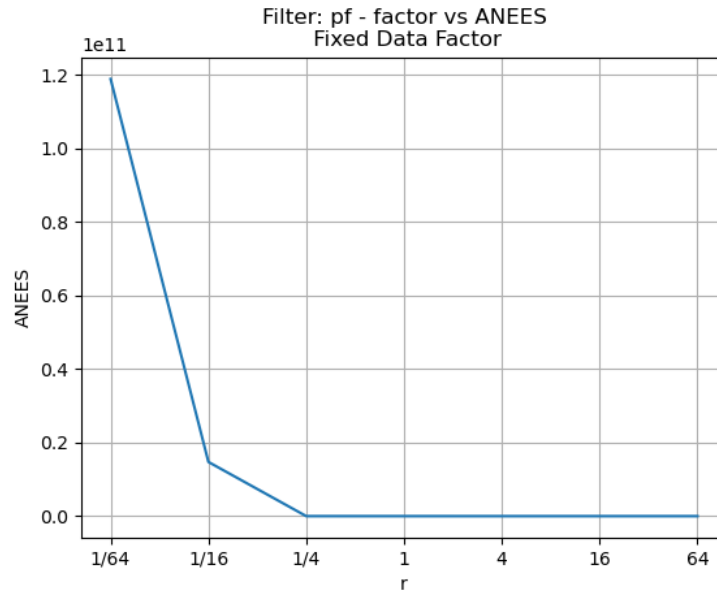


FIGURA 8. Valor medio del ANEES en 10 corridas para cada valor de  $r$  con factores de generación de datos fijos.

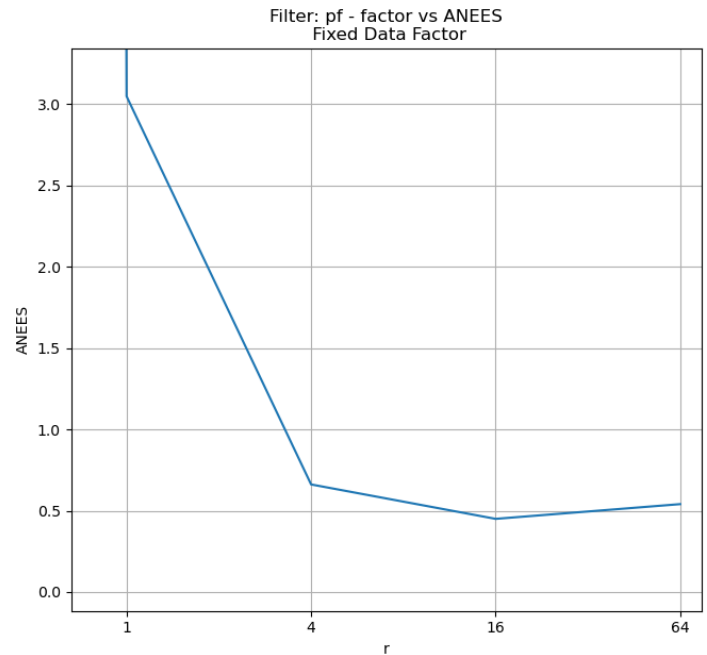
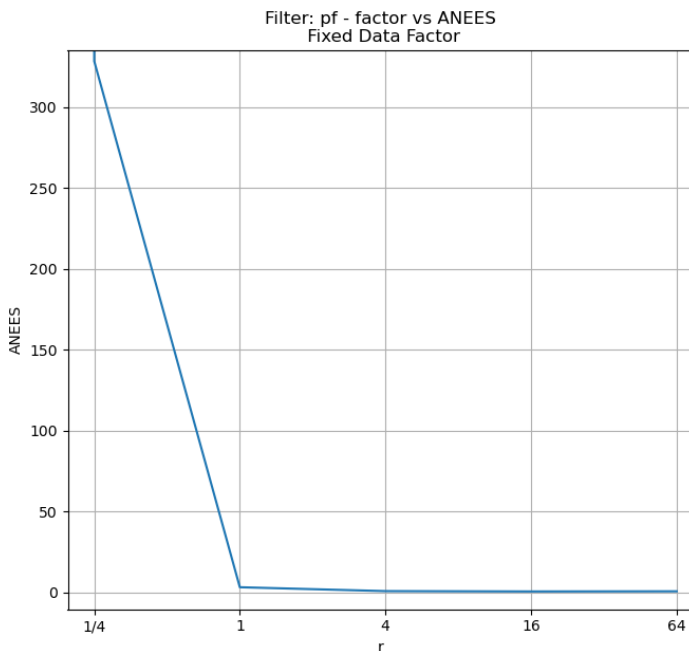


FIGURA 9. Valor medio del ANEES en 10 corridas para los valores de  $r$  entre 1/4 y 64 con factores de generación de datos fijos.

Podemos obtener conclusiones similares a las mencionadas en el EKF. Sin embargo, en este caso sobreestimar o subestimar la varianza de los procesos de evolución del sistema afecta tanto a la estima del estado del mismo como a la varianza de la estima, con lo cual el PF es más dependiente en el modelo de ruido que EKF.

d) En este ejercicio repetimos las simulaciones realizadas en el ítem b variando además la cantidad de partículas en  $p = [20, 50, 500]$ . En la Fig. 10 se muestra la media del MPE obtenido en las 10 ejecuciones para cada valor de  $r$  y  $p$ , mientras que en la Fig. 11 se muestra la media del ANEES y en la Fig. 12.

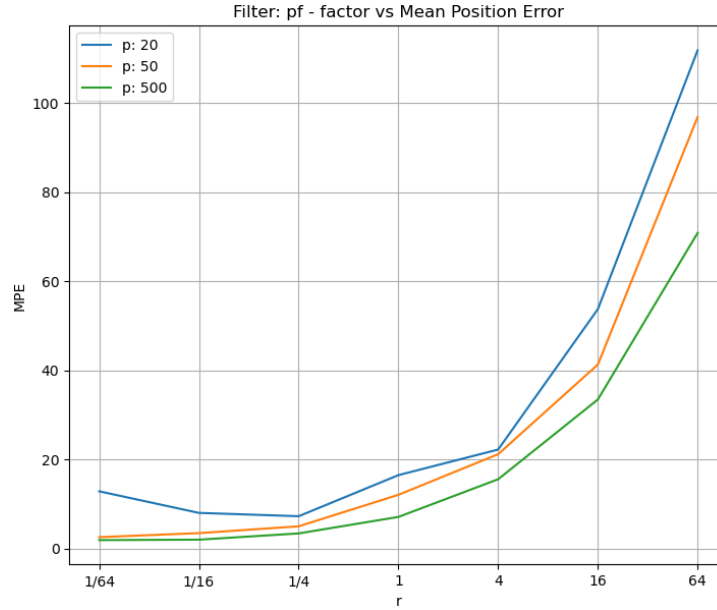


FIGURA 10. Valor medio del MPE en 10 corridas para cada valor de  $r$  variando la cantidad de partículas.

Podemos notar que, para una cantidad de partículas fija, el MPE aumenta al aumentar las varianzas, como habíamos observado en el ítem b). Asimismo, para un mismo valor de varianzas (mismo  $r$ ) el MPE es menor al aumentar la cantidad de partículas.

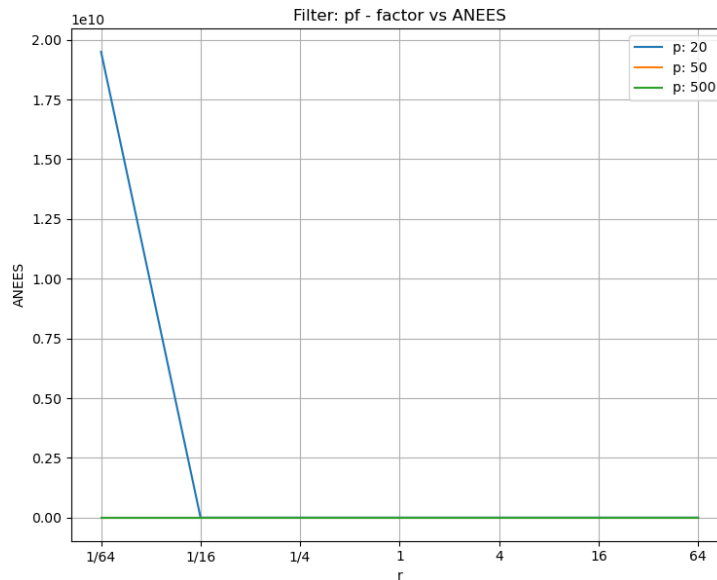


FIGURA 11. Valor medio del ANEES en 10 corridas para cada valor de  $r$  variando la cantidad de partículas.



Por otro lado, para una cantidad de partículas fija, podemos observar en el ANEES el mismo comportamiento que mostraba en el ítem anterior: cuando las varianzas empleadas en el filtro son mayores a las varianzas de la generación de datos (casos con  $r > 1$ ) el ANEES es pequeño, mientras que al emplear varianzas en el filtro menores a las varianzas de la generación de datos (casos con  $r < 1$ ) el ANEES es mayor. También ocurre que para valores pequeños de  $r$  la matriz de covarianza resulta en general no invertible y los resultados obtenidos para el ANEES son extremadamente altos. Además, al disminuir la cantidad de partículas este fenómeno continua ocurriendo para mayores valores de  $r$ . Asimismo, el aumento de partículas disminuyó el ANEES para todos los valores de  $r$ .

En conclusión, el aumento de la cantidad de partículas permite obtener un menor error de estima (menor MPE) y una mejor estima de la varianza de la misma.

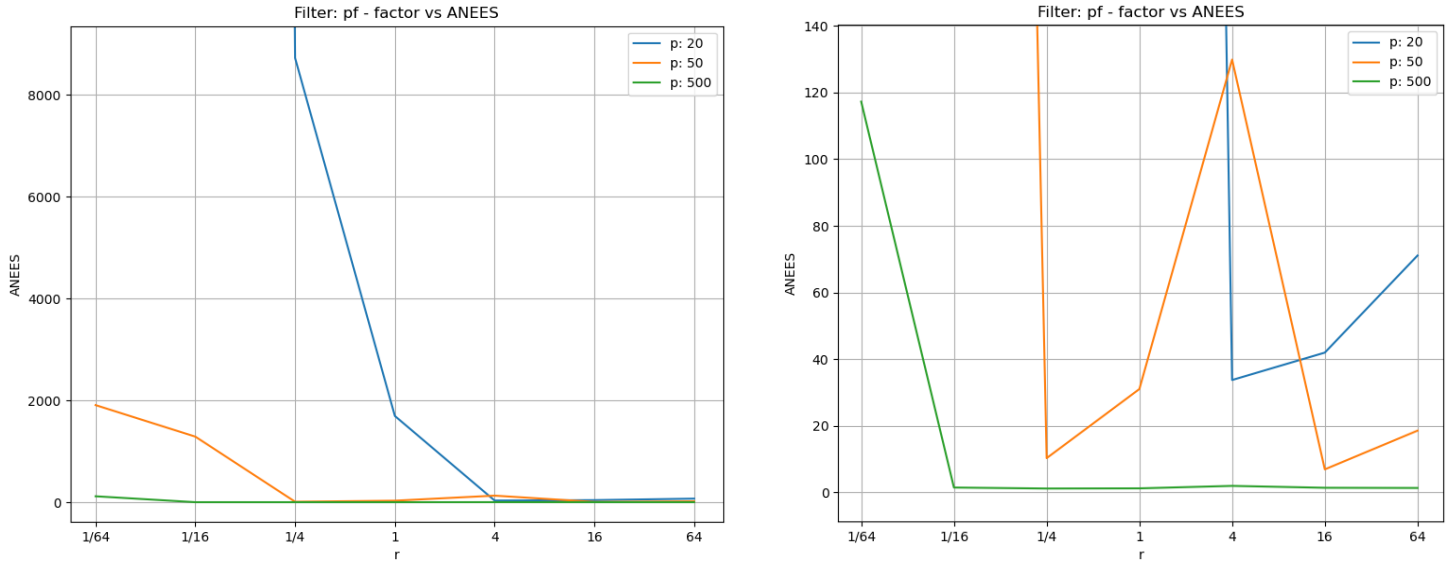


FIGURA 12. Valor medio del ANEES en 10 corridas para cada valor de  $r$  variando la cantidad de partículas.