A Direct Method for Checking Overlap of Two Hyperellipsoids

Igor Gilitschenski and **Uwe D. Hanebeck**
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute of Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
gilitschenski@kit.edu, uwe.hanebeck@ieee.org

***Abstract*—In this work, we propose a method for checking whether two arbitrary-dimensional hyperellipsoids overlap without making use of any optimization or root-finding methods. This is achieved by formulating an overlap condition as a polynomial root counting problem, which can be solved directly. The addressed challenges involve the inversion of a polynomial matrix using a direct method. The proposed approach extends one of our earlier results, which was restricted to certain combinations of ellipsoids and yields a fixed run-time for a fixed problem dimensionality. Thus, for the first time, an algorithm for checking overlap of arbitrary hyperellipsoids is proposed that can be evaluated in closed form. That is, in the absence of cut-off errors, the proposed method yields an exact result after a finite number of steps.**

***Keywords*—Hyperellipsoid overlap, Sturm theorem, Leverrier algorithm**

I. Introduction

Checking the intersection of hyperellipsoids is of interest in many scenarios, because hyperellipsoids can be used to account for uncertainty or as an easily computable approximation for other objects. Thus, a test for overlap is motivated by a broad range of applications and of particular use in handling data validation and fault-detection problems in data fusion scenarios where hyperellipsoids are used to represent uncertainty bounds. Furthermore, algorithms addressing this challenge can also be applied to on-line collision avoidance and probabilistic path planning, e.g., in robotics. Testing whether hyperellipsoids overlap is also used in the field of computer graphics (e.g., game development and CAD systems), in many physical simulations, and in statistics (where covariance ellipsoids of the normal distribution generalize the concept of σ-bounds to higher dimensions).

This seemingly simple problem poses a number of interesting challenges. On the one hand, it is of particular interest to come up with computationally efficient and robust algorithms. On the other hand, a thorough investigation of the entire problem structure promises to yield interesting theoretical results. Thus, it is not surprising that a broad number of publications has considered hyperellipsoid overlap. First, hyperellipsoid overlap tests in the context of fault detection were discussed in [1], [2], [3]. An algebraic condition for the separation of two ellipsoids in the Euclidean space was presented in [4]. In a similar setup, [5] proposed a test for ellipsoidal intersection, which is based on observing eigenvalue behaviour. Furthermore, several ellipsoid-based continuous collision detection algorithms have been proposed in [6], [7], [8]. All of these recently discussed algorithms usually restrict their consideration to the two- or three-dimensional case, because their motivation stems from computer graphics and robotic applications. In higher dimensions, such tests are usually based on optimization or root-finding methods. Such an algorithm is used in the ellipsoidal toolbox [9].

Furthermore, an efficient testing approach for arbitrary dimensions was discussed in [10] and later improved in [11]. The problem representation in these works was used in [12], where we proposed a novel method for testing whether two n-dimensional hyperellipsoids overlap. The algorithm was applied to a fault-detection scheme in Kalman filters. Even though our newly proposed method had a higher computational complexity, it offered a surprising result, because we could show that checking whether two hyperellipsoids overlap did not require the use of optimization or root-finding techniques even in higher dimensions. That is, the limitations imposed by the Abel-Ruffini theorem (stating that there is no general algebraic solution to polynomial equations of degrees higher than 4), which was independently proven and investigated by Galois, do not present an obstacle to a closed-form check of overlap in an arbitrary-dimensional scenario.

As in [11], our proposed method was based on two steps. In the first step, we obtained the coefficients of a polynomial characterizing the overlap of the ellipsoids using Leverriere algorithm. This differs from [11], where the polynomial (or its derivative) was obtained using simultaneous diagonalization (which requires root-finding for eigenvalue computations). Then, in the second step, a polynomial variant of the Euclidean algorithm was used in order to perform polynomial root counting according to Sturm's theorem. This also differs from [11], where it was proposed to use either convex optimization on the polynomial itself or a bisection method on its derivative.

Unfortunately, the approach in [12] did not present a general algorithm capable of checking overlap for arbitrary pairs of hyperellipsoids. This was due to the fact that this approach assumed the difference of the matrices describing the shape of the hyperellipsoids to be invertible. Thus, it failed for certain combinations, e.g., when both ellipsoids only differed in the position of their centers.

In this paper, we address these limitations and extend our earlier method by contributing a general algorithm for checking the overlap of two hyperellipsoids. The proposed condition is based on interpreting the problem as a polynomial root counting problem on a compact interval. This can be performed using

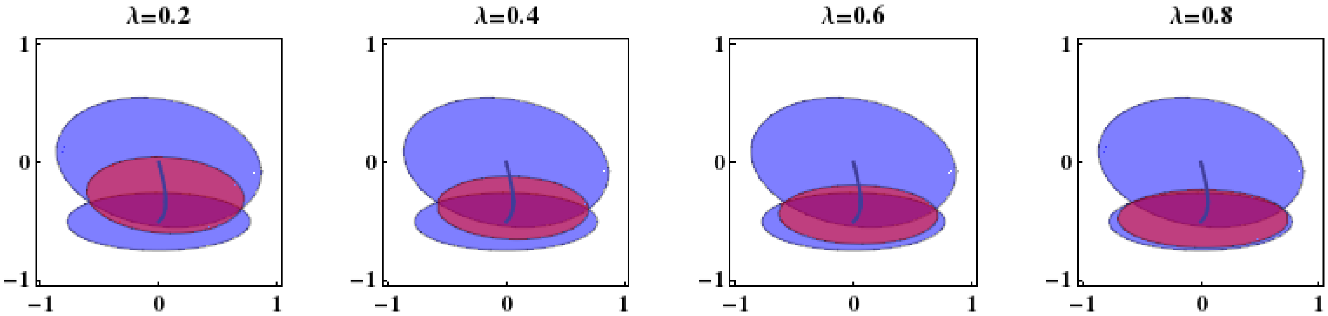

Figure 1: Checking whether the blue ellipsoids intersect can be thought of as checking whether the red ellipsoid $\mathcal{E}(\mathbf{E}_\lambda, \underline{m}_\lambda)$ disappears for some $\lambda \in (0, 1)$. The arc connecting the centers of the blue ellipsoids represents $\underline{m}_\lambda$ for $\lambda \in (0, 1)$.

Sturm's theorem that provides a method for polynomial root counting based on the Euclidean algorithm. A new test for overlap is proposed by adapting a modification of the Leverriere algorithm for computing the resolvent of a (possibly not regular) matrix and its computational complexity is discussed.

The remainder of this paper is structured as follows. In Sec. II, a modified version of the Leverrier algorithm is revisited, which was originally applied in the context of singular systems. It will be used as a key component in our newly proposed algorithm for computing the polynomial of interest. The condition for checking overlap of hyperellipsoids and its derivation is reviewed in Sec. III from an hyperellipsoid representation based on symmetric positive definite matrices. In Sec. IV, we present and discuss the algorithm for checking overlap, which does not require the use of numerical optimization techniques. The work is concluded in Sec. V.

II. A Modified Leverrier Algorithm

The Leverrier algorithm [13] is used to compute the resolvent of a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, which is defined as $(\lambda\mathbf{I} - \mathbf{A})^{-1}$ (where $\lambda \in \mathbb{R}$ and $\mathbf{I}$ is the $n \times n$ identity matrix). This is performed by simultaneously computing the determinant and the adjoint of $(\lambda\mathbf{I} - \mathbf{A})$ and then using

$$(\lambda\mathbf{I} - \mathbf{A})^{-1} = \frac{\operatorname{adj}(\lambda\mathbf{I} - \mathbf{A})}{\det(\lambda\mathbf{I} - \mathbf{A})} \ .$$

For a regular matrix $\mathbf{M}$, this algorithm can be simply adapted to compute $(\lambda\mathbf{M} - \mathbf{A})^{-1}$, by computing $(\lambda\mathbf{I} - \mathbf{A}\mathbf{M}^{-1})^{-1}$ first. To evaluate the general condition outlined in this paper, a more general algorithm is needed which considers cases when $\mathbf{M}$ is singular. This generalization of Leverrier algorithm was proposed in [14], where a scheme is derived to compute $\mathbf{R}(\lambda) := \operatorname{adj}(\lambda\mathbf{M} - \mathbf{A})$ and $q(\lambda) := \det(\lambda\mathbf{M} - \mathbf{A})$ for not necessarily regular $\mathbf{M}$. This is performed by

$$\mathbf{R}(\lambda) = \sum_{k=0}^{n-1} \mathbf{R}_{n-1,k}\lambda^k, \quad q(\lambda) = \sum_{k=0}^{n} q_{i,k} \ ,$$

where

$$\mathbf{R}_{i+1,k} = \begin{cases} -\mathbf{M}\mathbf{R}_{i,k-1} + q_{i+1,k}\mathbf{I} & \text{if } k = i+1 \\ -\mathbf{A}\mathbf{R}_{i,k} - \mathbf{M}\mathbf{R}_{i,k-1} + q_{i+1,k}\mathbf{I} & \text{if } k = 1, ..., i \\ -\mathbf{A}\mathbf{R}_{i,0} + q_{i+1,k}\mathbf{I} & \text{if } k = 0 \end{cases}$$

and

$$q_{i+1,k} = \begin{cases} \dfrac{1}{i+1}\operatorname{tr}(\mathbf{M}\mathbf{R}_{i,k-1}) & \text{if } k = i+1 \\ -\dfrac{1}{i+1}\operatorname{tr}(\mathbf{A}\mathbf{R}_{i,k} - \mathbf{M}\mathbf{R}_{i,k-1}) & \text{if } k = 1, ..., i \\ -\dfrac{1}{i+1}\operatorname{tr}(\mathbf{A}\mathbf{R}_{i,0}) & \text{if } k = 0 \end{cases}$$

with the base case $\mathbf{R}_{0,0} = \mathbf{I}$.

III. Proposed Condition

Several representations exist for describing an hyperellipsoid. A typical one is based on a symmetric positive definite matrix describing the shape and orientation of the hyperellipsoid and a vector describing its center. This representation turns out to be very convenient for deriving an overlap condition, which will be performed by investigating a convex combination of two inequalities each describing an ellipsoidal set.

Notation 1. *Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be symmetric positive definite and $\underline{v} \in \mathbb{R}^n$. Then, a hyperellipsoid is described using the notation*

$$\mathcal{E}(\mathbf{A}, \underline{v}) := \{\underline{x} \in \mathbb{R}^n : (\underline{x} - \underline{v})^\top \mathbf{A} (\underline{x} - \underline{v}) \leq 1\} \ .$$

Using this notation, the goal can be stated as checking whether $\mathcal{E}(\mathbf{A}, \underline{v}) \cap \mathcal{E}(\mathbf{B}, \underline{w}) = \emptyset$ holds. Developing a criterion for overlap is based on describing ellipsoids $\mathcal{E}_\lambda$ (with $\lambda \in [0, 1]$) that satisfy

$$(\mathcal{E}(\mathbf{A}, \underline{v}) \cap \mathcal{E}(\mathbf{B}, \underline{w})) \ \subseteq \ \mathcal{E}_\lambda \ \subseteq \ (\mathcal{E}(\mathbf{A}, \underline{v}) \cup \mathcal{E}(\mathbf{B}, \underline{w})) \quad (1)$$

for every $\lambda \in (0, 1)$. The set $\mathcal{E}_\lambda$ is characterized in the following proposition.

Proposition 1. *Consider symmetric positive definite matrices $\mathbf{A}$, $\mathbf{B} \in \mathbb{R}^{n \times n}$ and arbitrary vectors $\underline{v}, \underline{w} \in \mathbb{R}^n$. Define*

$$\mathbf{E}_\lambda := \lambda\mathbf{A} + (1-\lambda)\mathbf{B} \ , \quad \underline{m}_\lambda := \mathbf{E}_\lambda^{-1}(\lambda\mathbf{A}\underline{v} + (1-\lambda)\mathbf{B}\underline{w}) \ ,$$
$$K(\lambda) := 1 - \lambda\underline{v}^\top\mathbf{A}\underline{v} - (1-\lambda)\underline{w}^\top\mathbf{B}\underline{w} + \underline{m}_\lambda^\top\mathbf{E}_\lambda\underline{m}_\lambda \ .$$

Then for a fixed $\lambda \in (0, 1)$ the set

$$\mathcal{E}_\lambda := \{\underline{x} \in \mathbb{R}^n : (\underline{x} - \underline{m}_\lambda)^\top \mathbf{E}_\lambda (\underline{x} - \underline{m}_\lambda) \leq K(\lambda)\} \quad (2)$$

is either empty, consists of one element, or is itself a hyperellipsoid.

Proof: The matrix $\mathbf{E}_\lambda$ is positive definite, because it is a convex combination of positive definite matrices. Thus, (2) is not satisfied by any $\underline{x} \in \mathbb{R}^n$ if $K(\lambda) < 0$ and it is only satisfied by $\underline{x} = \underline{m}_\lambda$ if $K(\lambda) = 0$. For $K(\lambda) > 0$, we have

$$(\underline{x} - \underline{m}_\lambda)^\top \frac{1}{K(\lambda)} \mathbf{E}_\lambda (\underline{x} - \underline{m}_\lambda) \leq 1 \ ,$$

which is satisfied by a hyperellipsoidal subset of $\mathbb{R}^n$. ∎

The parameters $\mathbf{A}$, $\mathbf{B}$, $\underline{v}$, and $\underline{w}$ are suppressed in the notation $\mathcal{E}_\lambda$, as they are clear in the context. The next proposition gives some further insight on the relation between overlap of two hyperellipsoids and the set $\mathcal{E}_\lambda$. Examples for this newly resulting ellipsoid are shown in Fig. 1.

Proposition 2. *Consider two ellipsoids $\mathcal{E}(\mathbf{A}, \underline{v})$ and $\mathcal{E}(\mathbf{B}, \underline{w})$. Then* (1) *holds for all $\lambda \in (0, 1)$.*

Proof: We consider the inequality

$$\begin{aligned} 1 \geq\ & \lambda(\underline{x} - \underline{v})^\top \mathbf{A} (\underline{x} - \underline{v}) \\ & + (1-\lambda)(\underline{x} - \underline{w})^\top \mathbf{B} (\underline{x} - \underline{w}) \ . \end{aligned} \tag{3}$$

The first goal is to show

$$\mathcal{E}_\lambda = \{\underline{x} \in \mathbb{R}^n : \ \underline{x} \text{ satisfies (3)}\} \ .$$

By considering symmetry of $\mathbf{A}$, $\mathbf{B}$ and using the notation from the previous proposition, we get

$$\begin{aligned} 1 \leq\ & \lambda(\underline{x} - \underline{v})^\top \mathbf{A} (x - v) + (1-\lambda)(\underline{x} - \underline{w})^\top \mathbf{B} (\underline{x} - \underline{w}) \\ =\ & \underline{x}^\top [\lambda \mathbf{A} + (1-\lambda)\mathbf{B}] \underline{x} - 2\underline{x}^\top [\lambda(\mathbf{A}\underline{v}) + (1-\lambda)\mathbf{B}\underline{w}] \\ & + \lambda \underline{v}^\top \mathbf{A} \underline{v} + (1-\lambda) \underline{w}^\top \mathbf{B} \underline{w} \\ =\ & \underline{x}^\top \mathbf{E}_\lambda \underline{x} + 2\underline{x}^\top \mathbf{E}_\lambda \underline{m}_\lambda + \lambda\, \underline{v}^\top \mathbf{A} \underline{v} + (1-\lambda)\, \underline{w}^\top \mathbf{B} \underline{w} \ . \end{aligned}$$

Completing the square yields

$$\begin{aligned} & 1 + \underline{m}_\lambda^\top \mathbf{E}_\lambda \underline{m}_\lambda \\ & \quad \leq (\underline{x} - \underline{m}_\lambda)^\top \mathbf{E}_\lambda (\underline{x} - \underline{m}_\lambda) + \lambda \underline{v}^\top \mathbf{A} \underline{v} + (1-\lambda) \underline{w}^\top \mathbf{B} \underline{w} \ , \end{aligned}$$

which is equivalent to the inequality in (2).

Now, it is easily seen that at least one of the inequalities

$$(\underline{x} - \underline{v})^\top \mathbf{A} (\underline{x} - \underline{v}) \leq 1 \ , \qquad (\underline{x} - \underline{w})^\top \mathbf{B} (\underline{x} - \underline{w}) \leq 1 \tag{4}$$

holds for every $x \in \mathcal{E}_\lambda$. Thus, $\mathcal{E}_\lambda \subseteq (\mathcal{E}(\mathbf{A}, \underline{v}) \cup \mathcal{E}(\mathbf{B}, \underline{w}))$.

Consider now an arbitrary $\underline{x} \in \mathcal{E}(\mathbf{A}, \underline{v}) \cap \mathcal{E}(\mathbf{B}, \underline{w})$. For every such $\underline{x}$, both inequalities in (4) are satisfied. Thus, the inequality in (2) is satisfied and $(\mathcal{E}(\mathbf{A}, \underline{v}) \cap \mathcal{E}(\mathbf{B}, \underline{w})) \subseteq \mathcal{E}_\lambda$. ∎

Taking both previous Propositions into account, we now have some deeper insight into the function $K(\lambda)$ and how it relates to the overlapping.

Corollary 1. *The ellipsoids $\mathcal{E}(\mathbf{A}, \underline{v})$ and $\mathcal{E}(\mathbf{B}, \underline{w})$ share no common point if and only if there is a $\lambda \in (0, 1)$ with $K(\lambda) < 0$.*

In order to have a computational convenient way testing this condition, it is desirable to take a closer look at $K(\lambda)$, which is actually a polynomial. Plots of $K(\lambda)$ for different ellipsoid constellations are shown in Fig. 2. The following theorem establishes the convexity of $K(\lambda)$ on $(0, 1)$. It was originally shown in [10] in the context of discussing a polynomial directly related to $K(\lambda)$. Here, we provide a proof based on simultaneous diagonalization in order to give some insight into the problem structure.

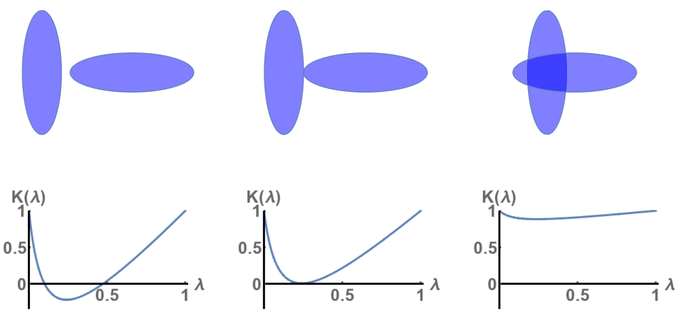

Figure 2: Plot of $K(\lambda)$ for different constellations of ellipsoids.

Proposition 3. *Consider two ellipsoids $\mathcal{E}(\mathbf{A}, \underline{v})$ and $\mathcal{E}(\mathbf{B}, \underline{w})$. Then, the corresponding function $K(\lambda)$ as defined in Proposition 1 is convex on $(0, 1)$.*

Proof: The function $K(\lambda)$ can be represented as

$$\begin{aligned} K(\lambda) = 1 - & \\ & (\underline{w} - \underline{v})^\top \left(\frac{1}{1-\lambda} \mathbf{B}^{-1} + \frac{1}{\lambda} \mathbf{A}^{-1} \right)^{-1} (\underline{w} - \underline{v}) \ . \end{aligned} \tag{5}$$

A discussion and a proof of this representation of $K(\lambda)$ are given in the Appendix. First, we consider the one-dimensional case. In this case for $K(\lambda)$ to be convex it is sufficient to show the convexity of

$$g(\lambda) = -\left(\frac{1}{1-\lambda} + \frac{b}{\lambda} \right)^{-1}$$

on $(0, 1)$ for all $b \in \mathbb{R}^+$. We can reformulate $g(\lambda)$ as

$$g(\lambda) = \frac{\lambda^2 - \lambda}{\lambda(1-b) + b} \ .$$

Its second derivative

$$g''(\lambda) = \frac{2b}{(b(1-\lambda) + \lambda)^3}$$

is positive for every $b \in \mathbb{R}^+$ and every $\lambda \in (0, 1)$. Thus, $g(\lambda)$ is convex on $(0, 1)$.

Now, we generalize this to higher dimensions. Using simultaneous diagonalization, $K(\lambda)$ can be represented as

$$K(\lambda) = 1 - (\underline{w} - \underline{v})^\top \mathbf{C}^\top \left(\frac{1}{1-\lambda} \mathbf{D}_1 + \frac{1}{\lambda} \mathbf{D}_2 \right)^{-1} \mathbf{C} (\underline{w} - \underline{v}) \ ,$$

where $\mathbf{D}_1$, $\mathbf{D}_2$ are diagonal matrices and

$$\mathbf{A}^{-1} = \mathbf{C}^{-1} \mathbf{D}_2 \mathbf{C}^{-\top} \ , \qquad \mathbf{B}^{-1} = \mathbf{C}^{-1} \mathbf{D}_1 \mathbf{C}^{-\top} \ .$$

Algorithm 1 Checking overlap of two Ellipsoids

```
procedure OVERLAPTEST(A, B, v, w)
    m_0 ← B w;
    m_1 ← (A v − B w);
    q_n ← getQ((A − B), −B, n, n);
    for k=0:(n-1) do
        R ← getR((A − B), −B, n − 1, k);
        q_k ← getQ((A − B), −B, n, k);
        K_k ← K_k + w^⊤ B^⊤ R B w;
        K_{k+1} ← K_{k+1} + 2 · (A v − B w)^⊤ R B w;
        K_{k+2} ← K_{k+2} + (A v − B w)^⊤ R (A v − B w);
    end for
    d(λ) ← polynomialGCD(K(λ), q(λ));
    K(λ) ← K(λ)/d(λ);
    q(x) ← q(λ)/d(λ);
    K(x) ← K(λ)
          +q(λ) [(w^⊤ B w − v^⊤ A v)λ + (1 − w^⊤ B w)];
    return Countroots(K(λ), 0, 1);
end procedure
```

$\mathbf{D}_1$ and $\mathbf{D}_2$ have only positive entries because $\mathbf{A}^{-1}$ and $\mathbf{B}^{-1}$ are positive definite. Thus, there exist $c_i, b_i \in \mathbb{R}^+$ such that $K(\lambda)$ can be written as

$$K(\lambda) = 1 + c_1 g(\lambda, b_1) + \ldots + c_n g(\lambda, b_n) \ ,$$

where $g(\lambda, b_i)$ is defined in the same way as $g(\cdot)$ with b replaced by b_i. This representation as a weighted sum of convex functions shows the convexity of $K(\lambda)$ and completes the proof. ∎

An immediate consequence of this theorem is the fact that $K(\lambda)$ has two distinct roots iff the hyperellipsoids do not overlap. This fact and the convexity of $K(\lambda)$ can both be used for implementing an efficient testing algorithm.

IV. ALGORITHMIC IMPLEMENTATION

According to the results of the previous section, it is sufficient to investigate $K(\lambda)$ on $(0,1)$ for checking overlap of two hyperellipsoids. There are at least two possible strategies for the algorithmic implementation. First, convex optimization can be used to find the minimum λ^* of $K(\lambda)$ (which is equivalent to finding the zero of $K'(\lambda)$). Checking the value $K(\lambda^*)$ yields the desired result. That is, the ellipsoids do not overlap for $K(\lambda^*) < 0$, they do overlap for $K(\lambda^*) \geq 0$, and they share only one common point for $K(\lambda^*) = 0$.

Second, the fact that $K(\lambda)$ is a polynomial makes a direct evaluation of the presented condition possible, because Sturm's theorem [15], [16] can be applied. This result from the field of algebraic geometry provides an algorithm for counting polynomial roots in a closed interval. In the following, we will describe the proposed algorithm by motivating and discussing the use of the modified Leverierre Algorithm and then describing the application of Sturm's theorem.

A. Proposed Algorithm

Similarly to our earlier approach [11], the overlap checking algorithm is divided into two parts. In the first part, we are interested in obtaining the coefficients of the polynomial $K(\lambda)$, which are used in the second part for polynomial root counting. This part will involve polynomial operations, such as polynomial division and thus, it is not sufficient to merely provide an evaluation method for $K(\lambda)$ (which would be computationally less demanding than the proposed method).

Algorithm 2 Counting Roots of a Polynomial

```
procedure COUNTROOTS(K(λ), λ_l, λ_r)
    p_0(λ) ← K(λ);
    p_1(λ) ← K'(λ);
    val_{l,0} ← sign(p_0(λ_l));
    val_{r,0} ← sign(p_0(λ_r));
    sc_l ← 0;
    sc_r ← 0;
    k ← 1;
    while polynomialRemainder(p_{k−1}, p_k) ≠ 0 do
        val_{l,k} ← sign(p_k(λ_l));
        val_{r,k} ← sign(p_k(λ_r));
        if val_{l,k} · val_{l,k−1} < 0 then
            sc_l ← sc_l + 1;
        end if
        if val_{r,k} · val_{r,k−1} < 0 then
            sc_r ← sc_r + 1;
        end if
        p_{k+1} ← polynomialRemainder(p_{k−1}, p_k);
        k ← k + 1;
    end while
    return sc_l − sc_r;
end procedure
```

An efficient method for obtaining the coefficients of $K(\lambda)$ was presented in [11], which uses simultaneous diagonalization and thus requires an eigenvalue computation. Due to the Abel-Ruffini theorem, there is no general algebraic solution for obtaining these eigenvalues. In order to uphold our claim of presenting a direct method checking ellipsoid overlap, we propose a different approach for obtaining the coefficients of $K(\lambda)$.

In the following, $K(\lambda)$ is represented as in Proposition 1 (the computational challenges discussed here would remain if we would have chosen another representation of $K(\lambda)$). In this situation, we have

$$\begin{aligned}&\underline{m}_\lambda^\top \mathbf{E}_\lambda \underline{m}_\lambda \\ &\quad = (\lambda \mathbf{A}\underline{v} + (1-\lambda)\mathbf{B}\underline{w})^\top \mathbf{E}_\lambda^{-1} (\lambda \mathbf{A}\underline{v} + (1-\lambda)\mathbf{B}\underline{w})^\top\end{aligned}$$

Thus, the main challenge is the computation of $\mathbf{E}_\lambda^{-1}$. If $(\mathbf{A}-\mathbf{B})$ is invertible, $\mathbf{E}_\lambda^{-1}$ can be reformulated as

$$\begin{aligned}\mathbf{E}_\lambda^{-1} &= (\lambda(\mathbf{A}-\mathbf{B}) + \mathbf{B})^{-1} \\ &= (\mathbf{A}-\mathbf{B})^{-1}(\lambda \mathbf{I} - \mathbf{B}(\mathbf{A}-\mathbf{B})^{-1})^{-1} \ .\end{aligned}$$

Here, Leverrier algorithm can be used directly for computing $(\lambda \mathbf{I} - \mathbf{B}(\mathbf{A}-\mathbf{B})^{-1})^{-1}$. Unfortunately, $(\mathbf{A}-\mathbf{B})$ can be singular. Thus, the modified version of Leverrier algorithm which was presented in Sec. II is applied for computing $\mathbf{E}_\lambda^{-1}$ directly.

This modified Leverrier algorithm is adapted to computing the factors of the polynomial $K(\lambda)$. The resulting procedure for the entire check for overlap is presented in Algorithm 1. That is, we compute the determinant and the adjoint of $\mathbf{E}_\lambda$ in order to compute its inverse. The functions getR and getQ are computed as described in the recursion schemes for $R_{i,k}$ and

$q_{i,k}$ in Sec. II, where the first two parameters take the role of $\mathbf{M}$ and $\mathbf{A}$ respectively. Both computations involve recursive calls and can be optimized by a precomputation before the `for`-loop.

The procedure `Countroots` returns the number of roots of $K(\lambda)$. The basic idea is applying Sturm's theorem, which provides a direct method for polynomial root counting. Thus, counting the number of real zeros of $K(\lambda)$ on $(0,1)$ is possible without making use of any optimization techniques. Instead, a polynomial version of the Euclidean algorithm has to be used in order to generate a so-called Sturm's sequence. This is the finite sequence of polynomials $p_0(\lambda), \ldots, p_L(\lambda)$ resulting from polynomial divisions within the Euclidean algorithm (thus, we can always guarantee $L \leq n$). Here, the result from the first step of the algorithm is used, because performing polynomial divisions requires the knowledge of the polynomial matrix coefficients. The resulting sequence can be used to count roots on an arbitrary interval (λ_l, λ_r) by comparing the number of sign changes in $p_0(\lambda_l), \ldots, p_L(\lambda_l)$ with the number of sign changes in $p_0(\lambda_l), \ldots, p_L(\lambda_l)$. The root counting algorithm is shown in Algorithm 2.

Finally, we observe that the entire overlap checking method yields an exact result in the absence of cut-off errors after a finite number of steps for any constellation of two arbitrary dimensional ellipsoids. First, the proposed modification of Leverriere algorithm yields an exact (in the above sense) result after executing a fixed number of summations, (matrix-) multiplications, and trace computations. Second, this is also true for the polynomial Euclidean algorithm, which is the basis for root counting based on Sturm's theorem.

B. Analysis and Discussion

The most expensive part in terms of computational complexity of the proposed method is the modified Leverierre algorithm. The values $q_{i,k}$ and the matrices $\mathbf{R}_{i,k}$ should be computed at the first call of `getQ` for all relevant value pairs (i,k). Otherwise, the recursive computation scheme from Sec. II would be invoked in every cycle of the `for`-loop. The entire recursion scheme has a computational complexity of $\mathcal{O}(n^2 \cdot \mathrm{M})$, where n denotes the dimension of the ellipsoids and M denotes the computational complexity of matrix multiplication (which is $\mathcal{O}(n^3)$ for the straightforward multiplication algorithm). This is the dominant part, because the polynomial variant of the Euclidean algorithm (in its straightforward version) is in $\mathcal{O}(n^2)$.

A precision analysis of the entire algorithm is somewhat more involved. However, there are some interesting results concerning the implementation of exact (in the sense that they assume the absence of cut-off errors) algorithms using floating-point numbers. In [17], it was established that a very general class of such algorithms can be adapted to work correctly using floating-point numbers. We are particularly interested in the polynomial GCD, which was investigated in [18], [19].

The algorithm in [11] computes the parameters of $K'(\lambda)$ using simultaneous diagonalization which is in $\mathcal{O}(\mathrm{M})$. Then $K'(\lambda)$ is checked using a bisection method, which has computational complexity in $\mathcal{O}\left(n \cdot \log\left(\varepsilon^{-1}\right)\right)$, where ε denotes the desired precision of the bisection. Thus, it provides a lower computational complexity, even if (in our new method) more general algorithms for polynomial matrix inversion would be used, such as those discussed in [20], [21]. However, in practical low-dimensional high-precision applications where $\log\left(\varepsilon^{-1}\right) \gg n$, it might be useful to replace the bisection by the proposed root-counting approach while computing the coefficients of $K(\lambda)$ using simultaneous diagonalization.

V. Conclusions

A simple condition for checking overlap of arbitrary-dimensional ellipsoids was discussed in this work. This condition is based on considering a polynomial that is convex on $(0,1)$. We have shown that counting roots in this interval is sufficient for checking overlap of two hyperellipsoids of arbitrary dimension. The entire method provided in this work is direct in the sense that it yields an exact result after a finite number of steps when numerical round-off errors are neglected. Thus, our surprising insight is that the Abel-Ruffini theorem does not present an obstacle towards checking ellipsoid overlap.

However, in its current state it is outperformed in complexity by methods involving optimization or approximate root finding techniques. Consequently, further research on polynomial matrix inversion is needed in order to obtain a direct method with a comparable computational complexity.

Acknowledgements

This work was partially supported by a grant from the German Research Foundation within the Research Training Group RTG 1194 "Self-organizing Sensor-Actuator-Networks".

Appendix

A representation of $K(\lambda)$ similar to (5) was originally used in [22]. Here, we provide a proof which is carried out in two steps. First, we prove this representation for the scalar case. Thus, we replace vectors and matrices $\underline{v}, \underline{w}, \mathbf{A}, \mathbf{B}$ by scalar values $v, w, A, B \in \mathbb{R}$. In the scalar case, $K(\lambda)$ can be simplified as

$$
\begin{aligned}
K(\lambda) =& 1 - \lambda v^2 A - (1-\lambda) w^2 B \\
& + (\lambda A v + (1-\lambda) B w)^2 \cdot \underbrace{(\lambda A + (1-\lambda) B)^{-1}}_{=:C^{-1}}
\end{aligned}
$$

After using C as the common denominator for the relevant terms and carrying out some simplifications, we obtain the desired result

$$
\begin{aligned}
K(\lambda) =& 1 - \frac{\lambda^2 v^2 A^2 + (1-\lambda)^2 w^2 B^2}{C} \\
& - \frac{\lambda(1-\lambda) A B (v^2 + w^2)}{C} + \frac{(\lambda A v + (1-\lambda) B w)^2}{C} \\
=& 1 - \frac{\lambda(1-\lambda) A B (v-w)^2}{C} \\
=& 1 - \frac{\lambda(1-\lambda) A B (v-w)^2}{(\lambda A + (1-\lambda) B)} \\
=& 1 - \frac{(v-w)^2}{((1-\lambda)^{-1} B^{-1} + \lambda^{-1} A^{-1})} \ .
\end{aligned}
$$

In the second step, this result is generalized to the n-dimensional case by reducing it to n scalar transformations. This reduction is carried out by applying the simultaneous diagonalization argument in the same way as in the proof of Proposition 3.

REFERENCES

[1] T. Kerr, "A Two Ellipsoid Overlap Test for Real Time Failure Detection and Isolation by Confidence Regions," in *1974 IEEE Conference on Decision and Control including the 13th Symposium on Adaptive Processes*, vol. 13. IEEE, 1974, pp. 735–742.

[2] T. H. Kerr, "Real-time Failure Detection: A Nonlinear Optimization Problem That Yields a Two-Ellipsoid Overlap Test," *Journal of Optimization Theory and Applications*, vol. 22, no. 4, pp. 509–536, Aug. 1977.

[3] T. Kerr, "Statistical Analysis of a Two-Ellipsoid Overlap Test for Real-Time Failure Detection," *IEEE Transactions on Automatic Control*, vol. 25, no. 4, pp. 762–773, Aug. 1980.

[4] W. Wang, J. Wang, and M.-S. Kim, "An Algebraic Condition for the Separation of Two Ellipsoids," *Computer Aided Geometric Design*, vol. 18, no. 6, pp. 531–539, Jul. 2001.

[5] S. Alfano and M. L. Greer, "Determining if Two Solid Ellipsoids Intersect," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 1, pp. 106–110, Jan. 2003.

[6] Y.-K. Choi, W. Wang, Y. Liu, and M.-S. Kim, "Continuous Collision Detection for Two Moving Elliptic Disks," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 213–224, Apr. 2006.

[7] X. Jia, Y.-K. Choi, B. Mourrain, and W. Wang, "An algebraic approach to continuous collision detection for ellipsoids," *Computer Aided Geometric Design*, vol. 28, no. 3, pp. 164–176, Mar. 2011.

[8] Y.-K. Choi, J.-W. Chang, W. Wang, M.-S. Kim, and G. Elber, "Continuous collision detection for ellipsoids." *IEEE transactions on visualization and computer graphics*, vol. 15, no. 2, pp. 311–24, Jan. 2009.

[9] A. A. Kurzhanskiy and P. Varaiya, *Ellipsoidal Toolbox Manual*, 2006-2008.

[10] J. W. Perram and M. Wertheim, "Statistical Mechanics of Hard Ellipsoids. I. Overlap Algorithm and the Contact Function," *Journal of Computational Physics*, vol. 58, no. 3, pp. 409–416, May 1985.

[11] U. D. Hanebeck, "Lokalisierung eines mobilen Roboters mittels effizienter Auswertung von Sensordaten und mengenbasierter Zustandsschätzung," Dissertation, TU München, Referent: G. Schmidt, Korreferent: E. D. Dickmanns, Fortschrittsberichte VDI, Reihe 8: Meß-, Steuerungs- und Regelungstechnik, Nr. 643, VDI Verlag, Düsseldorf, 1997, iSBN 3-18-364308-1.

[12] I. Gilitschenski and U. D. Hanebeck, "A Robust Computational Test for Overlap of Two Arbitrary-dimensional Ellipsoids in Fault-Detection of Kalman Filters," in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012.

[13] M. S. Fadali and A. Visioli, *Digital Control Engineering: Analysis and Design*. Academic Press, 2012.

[14] B. Mertzios, "Leverrier's Algorithm for Singular Systems," *IEEE Transactions on Automatic Control*, vol. 29, no. 7, pp. 652 – 653, Jul. 1984.

[15] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in Real Algebraic Geometry*, 2nd ed. Springer, 2006.

[16] M. Coste, *An Introduction to Semialgebraic Geometry*, 2002.

[17] S. Kiyoshi and M. Sweedler, "A Theory of Stabilizing Algebraic Algorithms," Mathematical Sciences Institute, Cornell University, Tech. Rep., 1985.

[18] P. Khungurn, "Shirayanagi-Sweedler Algebraic Algorithm Stabilization and Polynomial GCD Algorithms," Master's Thesis, Massachusetts Institute of Technology, 2007.

[19] P. Khungurn, H. Sekigawa, and K. Shirayanagi, "Minimum Converging Precision of the QR-Factorization Algorithm for Real Polynomial GCD," in *Proceedings of the 2007 international symposium on Symbolic and algebraic computation - ISSAC '07*. New York, New York, USA: ACM Press, Jul. 2007, p. 227.

[20] A. Storjohann, "On the Complexity of Inverting Integer and Polynomial Matrices," *Computational Complexity (accepted)*, 2008.

[21] C.-P. Jeannerod and G. Villard, "Essentially Optimal Computation of the Inverse of Generic Polynomial Matrices," *Journal of Complexity*, vol. 21, no. 1, pp. 72–86, Feb. 2005.

[22] L. Ros, A. Sabater, and F. Thomas, "An Ellipsoidal Calculus Based on Propagation and Fusion." *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, vol. 32, no. 4, pp. 430–42, Jan. 2002.